SAMPLE TAILORING OF 2167A DIDS FOR SOFTWARE-FIRST LIFE CYCLE

Document Number CDRL Sequence No. 01880

August 14, 1990

Prepared for:

Electronic Systems Division
Air Force Systems Command, USAF
Hanscom AFB, MA 07131-5000

Prepared by:

IBM Federal Sector Division
800 N. Frederick Ave.
Gaithersburg, Maryland 20879

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 14, 1990	Final

4. TITLE AND SUBTITLE
Sample Tailoring of 2167A DIDS for Software-First Life Cycle

5. FUNDING NUMBERS
C: F19628-88-D-0032

6. AUTHOR(S)
M. Blumberg

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
IBM Federal Sector Division
800 N. Frederick Avenue
Gaithersburg, MD 20879

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Electronic Systems Division
Air Force Systems Command, USAF
Hanscom AFB, MA 01731-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
CDRL Sequence No. 1880

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The DoD-STD-2167A military standard provides a framework of activities, work products, reviews, etc. which can be tailored by contractors to the specific requirements of a project. The STARS program began this tailoring process by developing section-by-section guidelines for tailoring data item description (DID) sections of DoD-STD-2167A CDRL items for use as Software-First Life Cycle (SFLC) CDRL items. This was possible since there is a one-to-one mapping between DoD-STD-2167A deliverable items and SFLC deliverable items.

The SFLC provides a dramatically different approach to systems development by integrating the use of rapid prototyping, software use, concurrent engineering, and other emerging technologies.

This document contains a description of a proposed set of deliverable items which will support the SFLC, a mapping of DoD-STD-2167A deliverable items to the proposed SFLC deliverable items, and tailoring guidelines on a section-by-section basis for DoD-STD-2167A deliverable items.

Keywords:

14. SUBJECT TERMS
STARS, Software-First Life Cycle, DoD-STD-2167A, rapid prototyping, Software reuse. (KR)

15. NUMBER OF PAGES
129

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Abstract

This document, IBM Contract Data Requirements List (CDRL) Sequence Number 1880, Sample Tailoring of 2167A DIDs for Software-First Life Cycle, is the last of five evolving CDRL items to be delivered as part of the STARS IBM R66 (IR66) task. The first document, CDRL item 1840, Proposed Software-First Life Cycle CDRLs - Preliminary, identified and described a proposed set of CDRL items for the Software-First Life Cycle (SFLC). The second document, CDRL item 1850, SFLC DIDs with 2167A Sections, added a mapping of DoD-STD-2167A Data Item Description (DID) sections to the proposed SFLC CDRL items. The third document, CDRL item 1860, Proposed Software-First Life Cycle CDRLs - Final, was a refinement of CDRL item 1840 and 1850. New material included descriptions of the interrelationships between system-level and component-level SFLC CDRL items (documents). The fourth document, CDRL item 1870, Proposed Software-First Life Cycle DIDs - Final, added general guidelines for tailoring DoD-STD-2167A CDRL items for use as SFLC CDRL items. The DID text for the DoD-STD-2167A CDRL items was also provided in appendices. This CDRL item provides tailoring guidelines on a section-by-section basis for DoD-STD-2167A CDRL items, so they can be customized to the specific requirements of the SFLC and associated technologies, e.g., prototyping, reusability, Ada software design, and object-oriented design.

The SFLC, which was defined in detail in CDRL item 1240, "Software-First Life Cycle - Final Definition", provides a dramatically different approach to systems development by integrating the use of rapid prototyping, reusable components, concurrent engineering, and other new and emerging technologies. This approach, which takes advantage of the benefits of each of the included technologies, has the potential for a substantial improvement in productivity, while increasing the quality and reusability of the developed system.

The SFLC consists of 5 phases: Preliminary System Analysis, System Architecture, Software Growing, Productization and Production, and System Operation and Support. This document describes the major software development related CDRL items (documents) that are required in support of the Preliminary System Analysis, System Architecture, and Software Growing phases; and the productization part of the Productization and Production phase. The CDRL items for the production part of the Productization and Production phase and the System Operation and Support phase are not included, since they are tied primarily to the distribution and maintenance of the operational system and the primary focus of this document is new software development.

This document presents the work of the IBM R66 task, performed by Maurice Blumberg, under the R increment of the STARS program. It is a continuation of the IBM Q15 task, performed by Maurice Blumberg and Mary Catherine Ward, which defined the SFLC.

Table of Contents

List of Illustrations

Introduction

Goals

One of the goals of the STARS program is to develop a system development process to support new and emerging technologies such as reusability, prototyping, object-oriented design (OOD), and Ada software development. This will "provide a basis for the development of very large, complex (and in particular, the so-called embedded, real time) systems on a predictable time scale far shorter and for a cost far less than is possible with today's software technology and acquisition process, and with a quality that far exceeds the norm." (Gree89) To meet the "system development process" goal, a STARS Software-First Life Cycle (SFLC) was developed and defined in detail in CDRL item 1240, Software-First Life Cycle - Final Definition (IBM1240).

The SFLC consists of 5 phases: Preliminary System Analysis, System Architecture, Software Growing, Productization and Production, and System Operation and Support. The SFLC is consistent with the "software-first" approach for new system acquisition, defined in Attachment 4 to AF Regulation 800-14 (STAR87), and with the software process model, defined in "A Spiral Model of Software Development and Enhancement" (Boeh88). In addition, while developing the SFLC, the experiences of several projects were studied, many current research efforts were evaluated, and various technical exchanges were conducted internal and external to IBM. Special emphasis was devoted to lessons learned from projects using traditional waterfall life cycles and projects with successful experiences in using new technologies such as rapid prototyping and reusability. These projects are discussed in CDRL item 1240, Software-First Life Cycle-Final Definition.

This document, IBM CDRL item 1880, Sample Tailoring of 2167A DIDs for Software-First Life Cycle, is the last of five evolving CDRL items to be delivered as part of the STARS IBM R66 (IR66) task. The first document, CDRL item 1840, Proposed Software-First Life Cycle CDRLs - Preliminary (IBM1840), identified and described a proposed set of CDRL items, which will support the SFLC (as defined in CDRL item 1240) in meeting the STARS goals. The second document, CDRL item 1850, SFLC DIDs with 2167A Sections (IBM1850), added a mapping of DoD-STD-2167A Data Item Description (DID) sections to the proposed SFLC CDRL items. The third document, CDRL item 1860, Proposed Software-First Life Cycle CDRLs - Final (IBM1860), was a refinement of CDRL item 1840 and 1850. New material included descriptions of the interrelationships between system-level and component-level SFLC CDRL items (documents). The fourth document, CDRL item C1870, Proposed Software-First Life Cycle DIDs - Final (IBM1870), added general guidelines for tailoring DoD-STD-2167A CDRL items for use as SFLC CDRL items. The DID text for the DoD-STD-2167A CDRL items was also provided in appendices.

CDRL item 1880 (C1880) is a refinement of previous CDRL items and includes tailoring guidelines on a section-by-section basis for DoD-STD-2167A CDRL items. This will allow these documents to be customized to the specific requirements of the SFLC and its associated technologies. Thus, rather then defining a whole new set of SFLC CDRL item DIDs which would totally replace the DoD-STD-2167A CDRL items, C1880 provides an approach to transitioning to a Software-First Life Cycle using the existing DoD-STD-2167A DIDs. This will enable a project to utilize the SFLC, while taking advantage of the large experience base in using DoD-STD-2167A DIDs. This is very consistent with the MIL-HDBK-287, A Tailoring Guide for DOD-STD-2167A, Defense System Software Development, which states that "DoD-STD-2167A is designed to be compatible with any software development model" (DODHDBK). In fact, paragraph 4.1.1 of the DoD-STD-2167A standard (DOD2167A) states that the software development activities "may

overlap and may be applied iteratively or recursively". C1880 will also provide guidance to future STARS efforts to define domain-specific SFLC CDRL items.

Scope

In order to keep the scope of the IR66 task focused on software development, this document describes only the software development-related CDRL items that are required in support of the Preliminary System Analysis, System Architecture, and Software Growing phases of the SFLC; and the productization part of the Productization and Production phase. The CDRL items for the production part of the Productization and Production phase and the System Operation and Support phase are not included, since they are tied primarily to the distribution and maintenance of the operational system and not to software development. Thus, the following workproducts (i.e., documents), identified in CDRL item 1240, Software-First Life Cycle-Final Definition, are not included in this document:

- System Engineering Plan
- Hardware Development Plan
- System Installation and Acceptance Plan
- Delivery Plan
- System Operation and Support Documents
- Trade Study Report

In keeping with the software development focus, CDRL items from the DoD-STD-2167A, Defense System Software Development (DoD2167A), were reviewed to determine the relationship to similar SFLC CDRL items, identified in this document. However, it is important to note that the SFLC has an impact on the entire system life cycle and, thus, other DoD standards would also be affected by implementing a software-first apprach. It is recommended that future studies evaluate the impact of the SFLC on other DoD standards such as:

- DoD-STD-480 - Configuration Control-Engineering Changes, Deviations, and Waivers
- DoD-STD-483 - Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs
- MIL-STD-490 - Specification Practices
- MIL-STD-499 - Engineering Management
- MIL-STD-1521- Technical Reviews and Audits for Systems, Equipments, and Computer Software

In CDRL item 1240, the following categories of work products were identified:

- internal - workproducts developed within a phase
- external - workproducts output from one phase and input to another phase
- informal - workproducts intermediate to the development of formal work products
- formal - workproducts following rigorous standards and subject to baseline control

For a given project, internal or informal work products would be developed using DIDs similar to external and formal work products. The unique characteristics of a specific project, and process tailoring, would determine what specific documents a customer would require to be contract deliverables, i.e., CDRL items. In order to provide maximum flexibility for process tailoring all SFLC documents, described in this document, are considered CDRL items, whether internal, external, informal, or formal work products.

Structure of Document

This document is divided into three major sections. The first section, "High Level View of the SFLC" on page 4, presents a high level description of the SFLC. The second section, "Summary of Proposed SFLC CDRL Items" on page 7, presents a summary description of the proposed SFLC CDRL items using multiple tables. The third section, "DoD-STD-2167A to SFLC Mapping" on page 15, provides a description of each SFLC CDRL item, plus a mapping of DoD-STD-2167A CDRL items to SFLC CDRL items, first at the CDRL item level and then at the section level. The appendices of this document contain the general tailoring guidelines and specific section-level guidelines for customizing each of the related DoD-STD-2167A CDRL item DIDs for use as SFLC CDRL items. The text of the DIDs for these DoD-STD-2167A CDRL items is also included with the tailoring guidelines.

High Level View of the SFLC

This section presents a refined version of the high level view of the SFLC, which was provided as part of CDRL item 1240 (IBM1240). It is included here to provide a perspective for the definition of SFLC CDRL items.

The SFLC consists of 5 phases, as shown in Figure 1 on page 5: Preliminary System Analysis, System Architecture, Software Growing, Productization and Production, and System Operation and Support. The phases are similar to those defined in the draft standard, "Software-First Development Standard for Systems-in-the-Large (Draft)" (STAR87). The SFLC, as defined in C1240, is a refinement of the draft standard into an "executable" life cycle which integrates such technologies as concurrent engineering, rapid prototyping, and reusability. By combining and integrating these technologies into a well-defined and highly concurrent process, the SFLC has the potential for a substantial improvement in productivity while at the same time, increasing the quality of the system. The SFLC incorporates many of the concepts defined in the "spiral model" of software development (Boeh88), e.g., process tailoring, use of prototyping, and incorporation of software risk management.

The SFLC, unlike more traditional waterfall life cycles, is not sequential and is not specification or document-driven. Evaluation of system capabilities and trade-offs do not occur against a set of documents, but rather against incremental prototypes which have been "rapidly" developed from reusable components. This is not to say that documents do not play a role, just a different role, i.e., they define enough detail to provide a sufficient level of control to the development effort; and they capture the results and record the direction of system development instead of completely determining it. System requirements and design documents are definitized after a full-capability prototype is developed rather than at the start of a project when system requirements are often not well-understood and are subject to change. This eliminates the need for fully elaborated and approved specifications and design documents as completion criteria for early requirements and design activities (Boeh88). Thus, the "excessive paper" production/change/maintenance cycle, required of the system developer, and the "excessive paper" review, required of the customer is eliminated.

The first phase of the life cycle is Preliminary Systems Analysis, which begins with the receipt of the desired operational capability for the system to be developed. The main purpose of this phase is to understand the system and users' requirements well enough to enable prototyping to begin in the next phase. The workproducts produced by this phase provide only a partial description of the system, and its operational capabilities, and are intended to provide sufficient detail to allow development of the initial system prototype. It is also in this phase that:

- the process definition is tailored to the unique requirements of the project
- the development environment is built
- high-risk items are identified, along with strategies for resolving them
- long-lead-time new technologies are identified to allow the required research and development efforts to be initiated

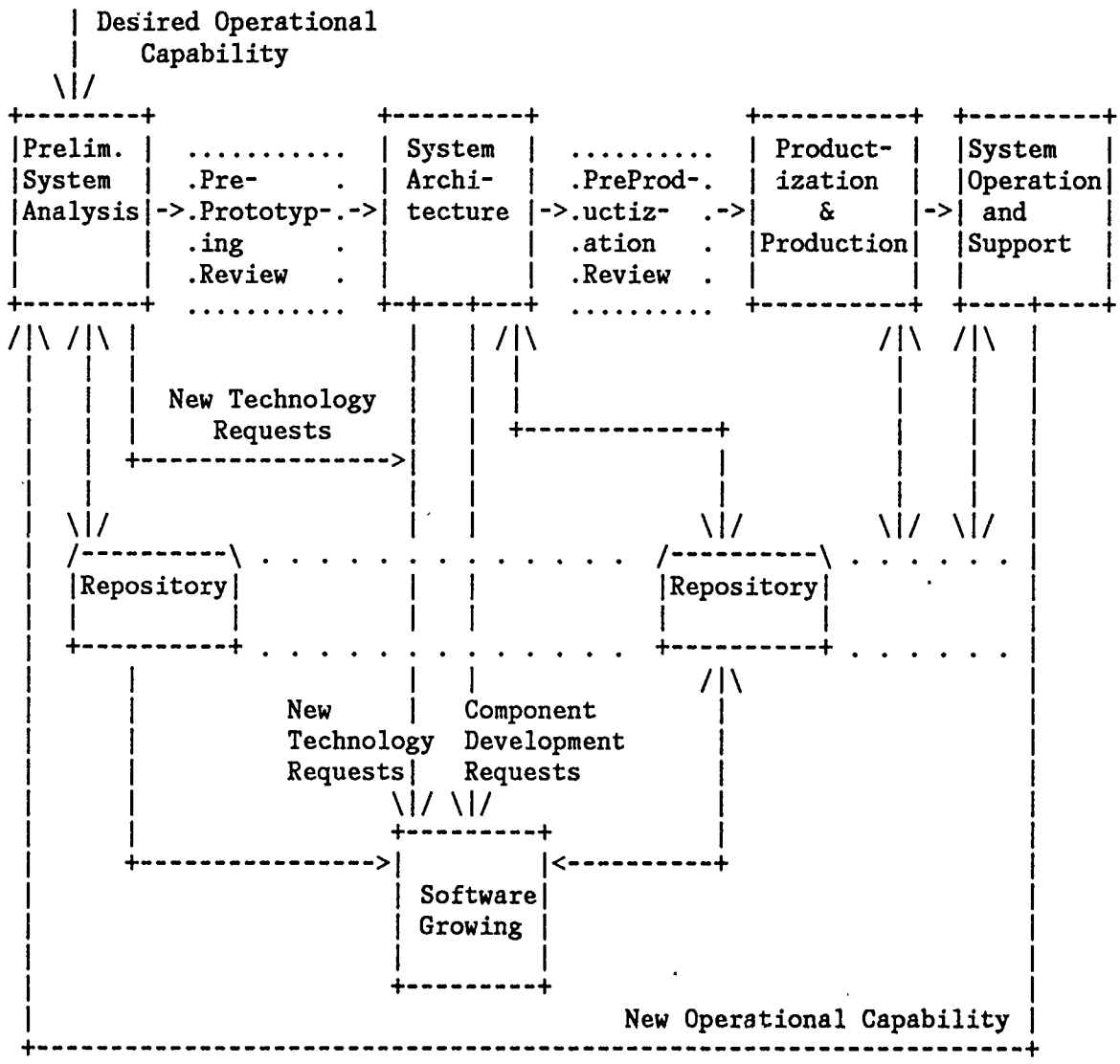

Figure 1. Software First Life Cycle - High Level View

The System Architecture phase is the driving phase of the life cycle. In this phase a full capability system prototype is evolved from a "rapidly" developed initial system prototype in order to address technological and programmatic risks early in the system life cycle and to allow early, effective involvement of the customer team in evaluating and refining system capabilities. The initial system prototype, and subsequent incremental system prototype builds, are assembled, to the greatest extent possible, from reusable components in the repository. It is during this phase that architecture trade analyses (e.g., hardware/software, performance/size) are performed. Alternative approaches are evaluated, as required, by building prototypes reflecting the alternative approach. When new software components or a new technology needs to be developed, requests are made to the Software Growing phase. Thus, as part of the process of evolving a full capability system prototype, incremental system prototypes are developed concurrently with new software components and new technologies. In addition, incremental system prototypes addressing alternative approaches of the same capabilitiy or addressing distinct capabilities are developed in parallel. These prototypes allow early involvement of the customer team, including planned users, in evaluating the system as it is being developed.

Within Software Growing new software components are developed, also using prototyping. Full capability component prototypes are productized and stored in the repository. Incremental component prototypes can also be stored in the repository to allow early use of software components in the assembly of a system prototype. It is also in this phase that new technologies, as needed, are researched and developed. When a new technology has been sufficiently defined, a new software component will be developed, as described previously, and stored in the repository.

Upon completion of the development of a full capability system prototype in the System Architecture phase, the prototype is productized in the Productization and Production phase. In this phase the system prototype, "a functionally executable software and hardware system (STAR87)", is converted into a productized system, "a fully defined, specified, developed, documented, tested, delivered and supported production quality software/hardware system (STAR87)".

Finally, in the System Operation and Support phase, the productized system is operated and maintained at the customer site. New operational requirements are identified for the next version of the system and these are input to the Systems Analysis phase. The SFLC is then repeated, as required, for a new version of the operational system.

Two major reviews, the Pre-Prototyping Review and the Pre-Productization Review are also shown in Figure 1. The Pre-Prototyping Review is performed after completion of the System Analysis phase to ensure that the system developer has a sufficient understanding of the system and users' requirements, to build a prototype. The review also ensures an understanding between the customer team and the system developer of the life cycle process tailoring, milestones, and deliverables to be followed by the project. The information that is reviewed, although not a complete definition, nevertheless, does provide a very concise definition of the system being developed and the capabilities of the initial prototype. The Pre-Productization Review is performed after completion of the System Architecture phase to ensure that the customer and the the system developer agree on the tasks necessary to productize the full capability prototype into an operational system.

Note:

In order to maintain diagram simplicity, feedback loops from a review or phase back to a previous phase are omitted from Figure 1. These feedback loops will occur when it is determined that further work is required in that phase.

Figure 1 also shows interaction with the repository in all phases of the life cycle, to indicate that more than just software is stored in the repository. In addition to software, such valuable information as lessons learned, results of domain analyses, and specifications will be stored and retrieved from the repository.

Summary of Proposed SFLC CDRL Items

This section presents a summary description of the Proposed SFLC CDRL Items. Nine SFLC documents have been identified.

- Software Development Plan
- System Description
- Prototype Capability Specification
- Prototype Design
- Software Test Document
- System Specification
- Software Design
- Component Specification
- Component Design

As mentioned previously, in order to provide maximum flexibility for process tailoring these documents are considered CDRL items, whether internal, external, informal, or formal documents. It should be understood that for a given project only a subset of the documents, listed as CDRL items in this document, could be actual CDRL items for that project.

The paragraphs below summarize the Proposed SFLC CDRL Items from multiple views, using various figures and tables. The first paragraph depicts the interrelationships between system-level and component-level CDRL items. The second paragraph describes how each CDRL item evolves during the execution of the SFLC phases. The third paragraph shows what CDRL items are input to and output from each phase of the SFLC. The fourth paragraph lists the external definitized CDRL items that describe the system. The fifth paragraph lists the external definitized CDRL items that describe the reusable software components developed during the execution of the SFLC. The last paragraph shows which CDRL items are internal, external, informal, or formal documents.

These tables reflect several refinements which were made to the SFLC workproducts, as defined in the CDRL item 1240. The Incremental Build Plan, Prototyping Plan, Usability Plan, and Productization Plan, which were part of the Project Plan workproduct, are now part of the Software Development Plan (SDP) CDRL item. The Environment Description workproduct has also been included as part of the SDP. The Software Test Document which was included as part of the System Test Plan (within the Project Plan) is now treated as a separate CDRL item. The New Technology Request and Component Development Request, which were identified as separate workproducts in C1240, will use the Prototype Capability Specification CDRL item format as the mechanism for requesting new technology or component development. The System Design workproduct has been replaced by the Software Design CDRL item.

CDRL Item Interrelationships

Figure 2 on page 8 shows the interrelationships between the various versions of system-level SFLC CDRL items and Figure 3 on page 9 shows the interrelationships between the various versions of

component-level SFLC CDRL items. These interrelationships show how documents evolve from previous versions and the interdependence of the documents. In the paragraphs below the documents will be further related to SFLC phases. For diagram simplicity, only major and direct interrelationships are provided. Thus, interrelationships between system and component-level CDRL items are not shown in the figures, as well as indirect interrelationships between the Software Development Plan and other CDRL items.

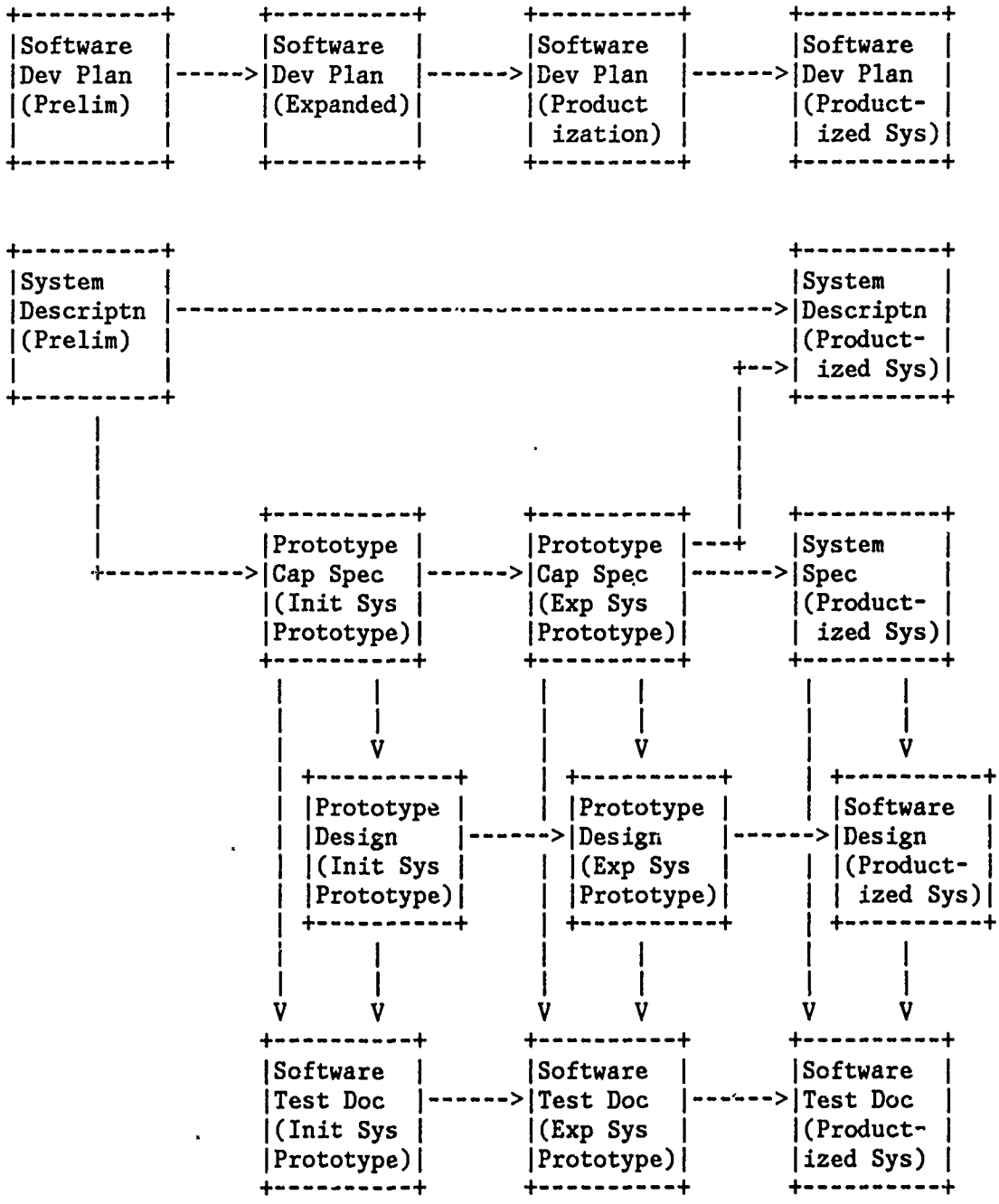

Figure 2. Interrelationship between System-Level SFLC CDRL Items

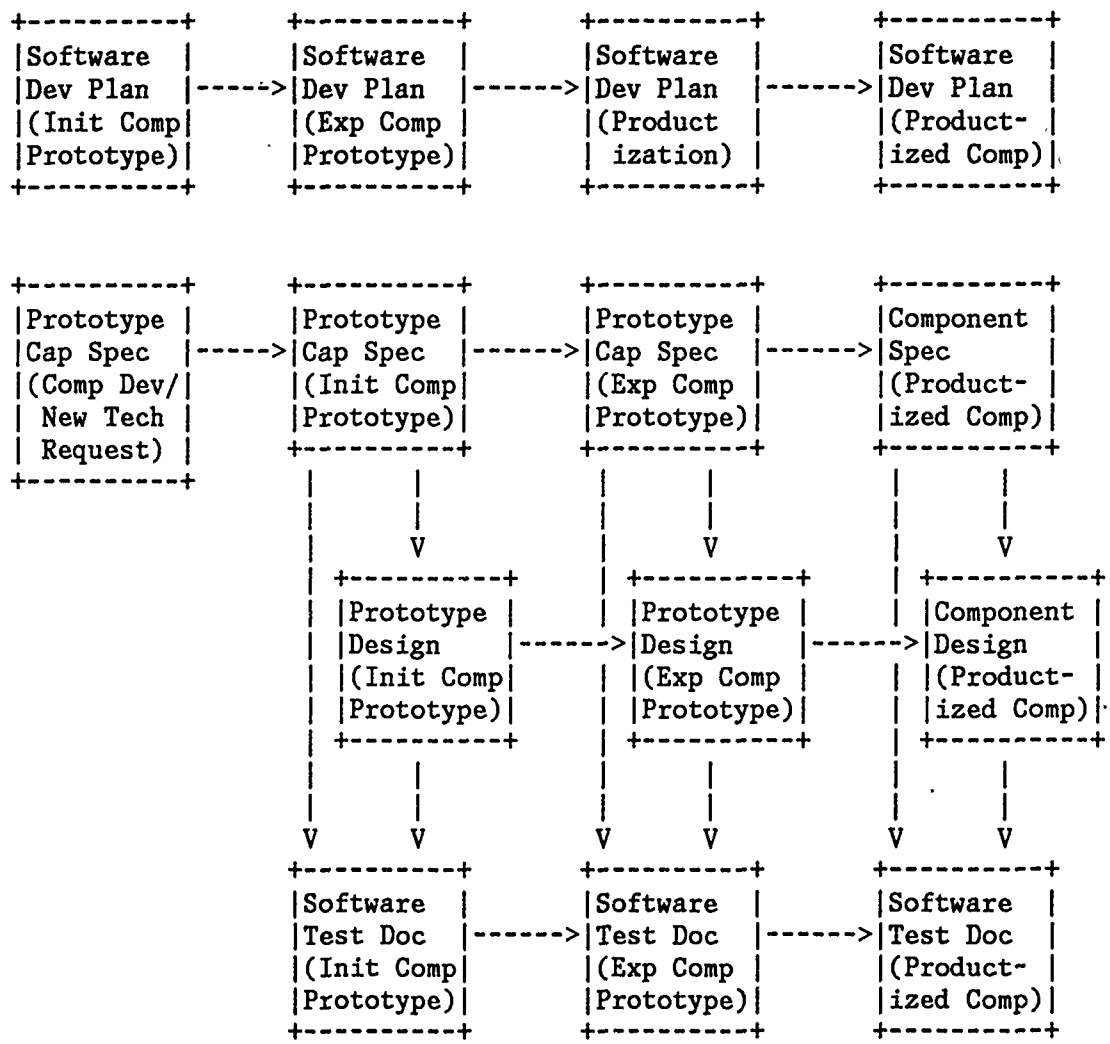

Figure 3. Interrelationship between Component-Level SFLC CDRL Items

CDRL Item Development Stages

The table below shows how CDRL items, both internal and external ones, are developed during the execution of the SFLC. For each CDRL item, informal (intermediate) versions of documents are listed to show how the CDRL item evolves within a phase.

Note: Since some CDRL items in the table below are internal ones, e.g., Prototype Design, they do not appear in the previous table, which only showed inputs to and outputs from the various SFLC phases.

CDRL Item	Preliminary System Analysis	System Architecture	Software Growing	Productization
Software Development Plan	Preliminary	• Expanded System Prototype • Productization	Component (All Versions)	Productized System
System Description	Preliminary	Definitized		
Prototype Capability Spec	• Initial System Prototype • New Technology Request	• Expanded System Prototype • Component Development • New Technology Request	• Initial Component Prototype • Expanded Component Prototype	
Prototype Design		• Initial System Prototype • Expanded System Prototype	• Initial Component Prototype • Expanded Component Prototype	
Software Test Document		• Initial System Prototype • Expanded System Prototype	• Initial Component Prototype • Expanded Component Prototype • Productized Component	Productized System
System Specification		Definitized		
Software Design				Definitized
Component Specification			Definitized	
Component Design			Definitized	

SFLC Phase Inputs and Outputs

The table below shows the CDRL items that are interchanged among the phases, including those CDRL items that are created by one phase and employed by other phases. Thus, the table does not show internal documents, i.e, those that are intermediate to the creation of an output document.

Phase	Inputs	Outputs
Preliminary System Analysis	• Statement of Work • Guidelines & Standards • Required Operational Capabilities	• System Description (Prelim) • Software Development Plan (Prelim) • Prototype Capability Specification(s) ▪ Initial System Prototype ▪ New Technology Request
System Architecture	• System Description (Prelim) • Software Development Plan (Prelim) • Prototype Capability Specification (Initial System Prototype)	• System Description (Definitized) • System Specification (Definitized) • Software Development Plan ▪ Expanded System Prototype ▪ Productization • Software Test Document (Expanded System Prototype) • Prototype Capability Specification(s) ▪ Component Development ▪ New Technology Request
Software Growing	• System Description (Preliminary) • Software Development Plan (Initial or Expanded System Prototype) • Software Test Document (Initial or Expanded System Prototype) • Prototype Capability Specification(s) ▪ Component Development ▪ New Technology Request	• Component Specification (Definitized) • Component Design (Definitized) • Software Development Plan (Component) • Software Test Document (Component)
Productization	• System Description (Definitized) • System Specification (Definitized) • Software Development Plan (Productization) • Software Test Document (Expanded System Prototype)	• Software Design (Definitized) • Software Development Plan (Productized System) • Software Test Document (Productized System)

Definitized External CDRL Items for the System

The table below shows the definitized external CDRL items which describe the "system", and are software development related :

Name	Output Phase
System Description	System Architecture
System Specification	System Architecture
Software Design	Productization and Production

Definitized External CDRL Items for Components

The table below shows the definitized external CDRL items which describe software "components".

Name	Output Phase
Component Specification(s)	Software Growing
Component Design(s)	Software Growing

CDRL Item Categorization

The table below shows which versions of CDRL items are internal, external, informal, or formal documents. Many of the CDRL items fall into multiple categories, e.g., many of the internal documents are also informal documents and many of the external documents are also formal documents. On a given project, the actual categories for each version of a document will be determined as part of the process tailoring.

CDRL Item	Internal	External	Informal	Formal
Software Development Plan	• Expanded System Prototype • Expanded Component Prototype	• Preliminary • Productization • Productized System • Initial Component Prototype • Component	• Preliminary • Expanded System Prototype • Initial Component Prototype • Expanded Component Prototype	• Productization • Productized System • Productized Component
System Description		• Preliminary • Definitized	Preliminary	Definitized
Prototype Capability Spec	• Expanded System Prototype • Expanded Component Prototype	• Initial System Prototype • New Technology Request • Initial Component Prototype	• Initial System Prototype • Expanded System Prototype • Initial Component Prototype • Expanded Component Prototype • New Technology Request	
Prototype Design	• Initial System Prototype • Expanded System Prototype • Initial Component Prototype • Expanded Component Prototype		• Initial System Prototype • Expanded System Prototype • Initial Component Prototype • Expanded Component Prototype	

CDRL Item	Internal	External	Informal	Formal
Software Test Document	• Initial System Prototype • Initial Component Prototype • Expanded Component Prototype	• Expanded System Prototype • Productized Component • Productized System	• Initial System Prototype • Expanded System Prototype • Initial Component Prototype • Expanded Component Prototype	• Productized Component • Productized System
System Specification		Definitized		Definitized
Software Design		Definitized		Definitized
Component Specification		Definitized		Definitized
Component Design		Definitized		Definitized

DoD-STD-2167A to SFLC Mapping

The subsections below contain mappings of related DoD-STD-2167A CDRL Items to SFLC CDRL Items. The first subsection provides a mappings of DoD-STD-2167A CDRL Items to SFLC CDRL Items. The next subsection provides a description of each SFLC CDRL item, and includes a mapping of DoD-STD-2167A CDRL Item sections to SFLC CDRL Items.

Although the mappings reflect a similarity in format and content between SFLC and DoD-STD-2167A documents, they are not intended to show a similarity in how they are used or when they are developed. On a standard DoD-STD-2167A project, definitized specifications and design documents are produced and authenticated prior to the start of the actual development of the software. Often, at that time, the system requirements are still not well-understood and are subject to change.

The SFLC, unlike a typical DoD-STD-2167A life cycle, is not specification or document-driven. The SFLC is prototype-driven, and is intended to significantly reduce the documentation overhead in the development process. Enough documentation is provided to control the process, but interaction with customers and end-users during development is dependent primarily upon prototype demonstrations and evaluations. Definitized versions of documents are not produced, or validated, until after a full-capablity system prototype has been developed. Thus, as mentioned previously, the "excessive paper" production/change/maintenance cycle, required of the system developer, and the "excessive paper" review, required of the customer is eliminated.

Mapping of DoD-STD-2167A CDRL Items to SFLC CDRL Items

This subsection describes a mapping of DoD-STD-2167A CDRL Items to SFLC CDRL Items. This mapping, which is shown in the table below, is not intended to show an inclusive one-to-one relationship, but is intended to indicate what DoD-STD-2167A CDRL items have parts (sections) which could be used in defining SFLC CDRL Items. The more specific mapping, at the section level, is provided in the next subsection.

SFLC CDRL Item	DoD-STD-2167A CDRL Item
Software Development Plan	Software Development Plan (SDP)
System Description	System/Segment Design Document (SSDD)
Prototype Capability Specification	System/Segment Specification (SSS)
Prototype Design	Software Design Document (SDD), Interface Design Document (IDD)
Software Test Document	Software Test Plan, Software Test Description, Software Test Report
System Specification	System/Segment Specification (SSS)
Software Design	Software Product Specification (SPS)
Component Specification	Software Requirements Specification (SRS), Interface Requirements Specification (IRS)
Component Design	Software Design Document (SDD), Interface Design Document (IDD)

Note: Those DoD-STD-2167A CDRL items that are basically support documents, and are not heavily dependent on the life cycle approach (e.g., software-first, waterfall, etc.), are not listed in the table. It is assumed that the DIDs for these CDRL items will not change. These include:

- Version Description Document
- Computer Systems Operator's Manual
- Software User's Manual
- Software Programmer's Manual
- Firmware Support Manual
- Computer Resources Integrated Support Manual

Mapping of DoD-STD-2167A Sections to SFLC CDRL Items

This subsection contains a high level definition of each SFLC CDRL item, including a mapping of DoD-STD-2167A CDRL item sections (as described in their associated DIDs) to SFLC CDRL items. Tailoring guidelines for customizing the DoD-STD-2167A CDRL items for use as SFLC CDRL items are provided in the appendices. This includes general guidelines and specific DoD-STD-2167A DID section-by-section tailoring guidelines for each SFLC CDRL item.

Note: The general tailoring guidelines, are intended to provide overall guidance for customizing the associated DoD-STD-2167A CDRL item DIDs for use in the SFLC. The specific tailoring guidelines provide more detailed guidance so that the associated DoD-STD-2167A CDRL item DIDs can be customized to the specific requirements of the SFLC and associated technologies, e.g., reusability, prototyping, object-oriented design, and Ada software development. Further tailoring of the DIDs will also be required, depending on such factors as size of the project, whether it is a precedented or unprecedented system, availability of reusable components, and user interface requirements (Boeh88).

The proposed SFLC CDRL items are described below in the order of the previous table. The template used to define each SFLC CDRL item includes the following:

- high-level description
- general content
- how it is developed

- whether it is an external or internal work product
- a mapping of related DoD-STD-2167A CDRL item section numbers to general content of SFLC CDRL item

To simplify the mappings below, and the specific tailoring guidelines in the appendices, Section 1, Scope and Section 2, Referenced Documents or Applicable Documents, which are common to all DoD-STD-2167A CDRL items, and the Notes and Appendixes sections, are not included. Thus, both the mappings and the specific tailoring guidelines for each DoD-STD-2167A CDRL item begin with section 3. In addition, to further simplify the mappings below, a DoD-STD-2167A CDRL item section number is used to represent the mapping of the entire section. Otherwise, subsection numbers are used.

1. SOFTWARE DEVELOPMENT PLAN

Type: Document.

Description:

The Software Development Plan includes all of the major plans, standards, and procedures necessary for completion of software development.

General Content:

- Software Management
  - Incremental Build Plan
  - Prototyping Plans
  - Productization Plan
  - Risk Management Plan
- Software Engineering
  - Organization and Resources (including Environment Description)
  - Standards and Procedures (including inspections)
  - Usability Plan
- Software Test Planning
  - Prototype Testing
  - Formal Qualification Testing
  - Product Evaluations
- Software Configuration Management
- Other Software Functions (includes Software Measures)

Development:

The Software Development Plan is incrementally expanded and updated, to reflect the current developmental stage, as the project life cycle is executed.

External/Internal:

The Software Development Plan is a working document and at different points in the life cycle, a snapshot of it is externalized.

Related DoD-STD-2167A Sections:

The table below shows a mapping of DoD-STD-2167A sections to the above general content, for this CDRL item. Following the table is an indented list of the DoD-STD-2167A sections/subsections included in the mapping.

2167A Section	SW Mgmt	SW Eng	SW Test Planning	SW CM	Other SW Fcns
Software Development Plan					
3. Software development management	X				
4. Software engineering		X			
5. Formal qualification testing			X		
6. Software product evaluation			X		
7. Software configuration management				X	
8. Other software development functions					X

The 2167A sections/subsections included in the above mapping are listed below:

Software Development Plan

3. Software development management
    3.1 Project organization and resources
    3.2 Schedule and milestones
    3.3 Risk management
    3.4 Security
    3.5 Interface with associate contractors
    3.6 Interface with software IV and V agent(s)
    3.7 Subcontractor management
    3.8 Formal reviews
    3.9 Software development library
    3.10 Corrective action process
    3.11 Problem/change report
4. Software engineering
    4.1 Organization and resources - software engineering
    4.2 Software standards and procedures
    4.3 Non-developmental software
5. Formal qualification testing
    5.1 Organization and resources - formal qualification testing
    5.2 Test approach/philosophy
    5.3 Test planning assumptions and constraints
6. Software product evaluation
    6.1 Organization and resources - software product evaluation
    6.2 Software product evaluations procedures and tools
    6.3 Subcontractor products
    6.4 Software product evaluation records
    6.5 Activity-dependent product evaluations
7. Software configuration management
    7.1 Organization and resources - configuration management
    7.2 Configuration identification
    7.3 Configuration control
    7.4 Configuration status accounting
    7.5 Configuration audits
    7.6 Preparation for specification authentication
    7.7 Configuration management major milestones
8.0 Other software development functions
    8.X (Function name, e.g., Software measures)

2. SYSTEM DESCRIPTION

Type: Document.

Description:

The System Description contains a high level description of the system mission, the operational need, and operational concept.

General Content:

- Mission Statement
  - Purpose of the system
  - Goals of the system
- Operational Need
  - Identification and evaluation of existing system deficiencies
  - Identification of additional operational capabilities
  - Prioritization of capabilities
  - Identification of constraints
- Operational Concept
  - Operational Environment Description
  - System Functional Overview
  - High Level System Architecture
- Rationale

Development:

- The Mission Statement is developed during the Preliminary System Analysis phase.
- A preliminary version of the Operational Need is developed during the Preliminary System Analysis Phase. It is definitized during the System Architecture Phase.
- A preliminary version of the Operational Concept is drafted in the Preliminary System Analysis phase. It is definitized during the System Architecture phase.

External/Internal:

The definitized System Description is an external delivery.

Related DoD-STD-2167A Sections:

The table below shows a mapping of DoD-STD-2167A sections to the above general content, for this CDRL item. Following the table is an indented list of the DoD-STD-2167A sections/subsections included in the mapping.

2167A Section	Mission Statement	Operational Need	Operational Concept	Rationale
System/Segment Design Document				
3. Operational concepts				
3.1 Mission	X			
3.2 Operational environment			X	
3.3 Support environment			X	
3.4 System architecture			X	
3.5 Operational scenarios			X	
4. System design			X	
5. Processing resources			X	
6. Quality factor compliance			X	
7.0 Requirements traceability			X	

The 2167A section/subsections included in the above mapping are listed below:

System/Segment Design Document

3. Operational concepts
   - 3.1 Mission
   - 3.2 Operational environment
   - 3.3 Support environment
   - 3.4 System architecture
   - 3.5 Operational scenarios
4. System design
   - 4.1 HWCI identification
   - 4.2 CSCI identification
   - 4.3 Manual operations identification
   - 4.4 Internal interfaces
5. Processing resources
   - 5.X (Processing resource name and project-unique identifier)
6. Quality factor compliance
7. Requirements traceability

3. PROTOTYPE CAPABILITY SPECIFICATION

Type: Document.

Description:

The Prototype Capability Specification details the capabilities of a system prototype, component prototype, or new technology to be developed.

General Content:

- Previous Capabilities/Interfaces (if applicable)
- New Capabilities/Interfaces
- Identification of Reusable Components
- Identification of Standard Interfaces
- Rationale

Development:

- A preliminary Prototype Capability Specification, for the development of an initial system prototype, is initiated from the Preliminary System Analysis phase to the System Architecture phase.

- A Prototype Capability Specification, requesting development of a software component, can be initiated from any phase, but most requests will be initiated from the System Architecture phase to the Software Growing phase.
- A Prototype Capability Specification, requesting development of a new technology, can be initiated from any phase, but most requests will be initiated from the Preliminary System Analysis, System Architecture, or Software Growing phases.

External/Internal:

Prototype Capability Specifications can be internal or external documents, as follows:

- A preliminary Prototype Capability Specification, for an initial system prototype or for new component development, is an external work product of an SFLC phase.
- Expanded Prototype Capability Specifications, for system or component prototypes, are informal internal work products.
- A Prototype Capability Specification, for a new technology, can be an internal or external work product.

Related DoD-STD-2167A Sections:

The table below shows a mapping of DoD-STD-2167A sections to the above general content, for this CDRL item. Following the table is an indented list of the DoD-STD-2167A sections/subsections included in the mapping.

2167A Section	Previous Capab./ Interfaces	New Capab./ Interfaces	Ident. of Reusable Comp.	Ident. of Std Interfaces	Rationale
System/Segment Specification					
3. System requirements					
3.1 Definition	X	X			
3.2 Characteristics	X	X	X	X	
3.3 Design and Construction	X	X	X		
3.4 Documentation					
3.5 Logistics					
3.6 Personnel and training					
3.7 Characteristics of subordinate elements	X	X			
3.8 Precedence					
3.9 Qualification					
3.10 Standard Sample					
3.11 Preproduction sample, periodic production sample, pilot, or					
4. Quality assurance provisions					

The 2167A section/subsections included in the above mapping are listed below:

System/Segment Specification

3. System requirements
    3.1 Definition
    3.2 Characteristics
    3.3 Design and Construction
    3.4 Documentation
    3.5 Logistics
    3.6 Personnel and training
    3.7 Characteristics of subordinate elements
    3.8 Precedence
    3.9 Qualification
    3.10 Standard Sample
    3.11 Preproduction sample, periodic production sample, pilot, or pilot lot
4. Quality assurance provisions
    4.1 Responsibility for inspection
    4.2 Special tests and examinations
    4.3 Requirements cross reference

4. PROTOTYPE DESIGN

Type: Document.

Description:

The Prototype Design details the implementation of a system or component prototype to be developed. (It may include the design of lower level reusable components which are part of this component.)

General Content:

- Architecture
- Design
- Data
- Interfaces
- Rationale

Development:

A Prototype Design is developed for each prototype under development.

External/Internal:

All Prototype Designs are informal, internal work products.

Related DoD-STD-2167A Sections:

The table below shows a mapping of DoD-STD-2167A sections to the above general content, for this CDRL item. Following the table is an indented list of the DoD-STD-2167A sections/subsections included in the mapping.

2167A Section	Architec-ture	Design	Data	Interfaces	Rationale
Software Design Document					
3. Preliminary Design	X				
4. Detailed Design		X			
5. CSCI (prototype) data			X		
6. CSCI (prototype) data files			X		
7. Requirements traceability		X			
Interface Design Document					
3. Interface Design				X	

The 2167A section/subsections included in the above mapping are listed below:

- Software Design Document
  - 3. Preliminary Design
    - 3.1 CSCI (prototype) overview
    - 3.2 CSCI (prototype) design description
  - 4. Detailed Design
    - 4.X (CSC (prototype) name and project unique identifier)
  - 5. CSCI (prototype) data
  - 6. CSCI (prototype) data files
    - 6.1 Data file to CSC/CSU (prototype) cross reference
    - 6.X (Data file name and project unique identifier)
  - 7. Requirements traceability
- Interface Design Document
  - 3. Interface Design
    - 3.1 Interface diagrams
    - 3.X (Interface name and project unique identifier)

5. SOFTWARE TEST DOCUMENT

Type: Document.

Description:

A Software Test Document is needed for each system and component prototype, as well as for the productized system software and each productized software component.

General Content:

- Test Planning
- Test Description
- Test Results

Development:

Software Test Documents are developed and refined in the System Architecture and Software Growing phases.

External/Internal:

Software Test Documents can be internal or external work products.

Related DoD-STD-2167A Sections:

The table below shows a mapping of DoD-STD-2167A sections to the above general content, for this CDRL item. Following the table is an indented list of the DoD-STD-2167A sections/subsections included in the mapping.

2167A Section	Test Planning	Test Description	Test Results
Software Test Plan			
3. Software test environment	X		
4. Formal qualification test identification	X		
5. Data recording, reduction, and analysis	X		
Software Test Description			
3. Formal qualification test preparations		X	
4. Formal qualification test descriptions		X	
Software Test Report			
3. Test overview			X
4. Test results			X
5. CSCI evaluation and recommendations			X

The 2167A section/subsections included in the above mapping are listed below:

- Software Test Plan
  - 3. Software test environment
    - 3.1 Software items
    - 3.2 Hardware and firmware items
    - 3.3 Proprietary nature
    - 3.4 Installation testing and control
  - 4. Formal qualification test identification
    - 4.X (CSCI name and project unique identifier)
  - 5. Data recording, reduction, and analysis
- Software Test Description
  - 3. Formal qualification test preparations
    - 3.X (Test name and project unique identifier)
  - 4. Formal qualification test descriptions
    - 4.X (Test name and project unique identifier)
- Software Test Report
  - 3. Test overview
  - 4. Test results
  - 5. CSCI evaluation and recommendations

6. SYSTEM SPECIFICATION

Type: Document.

Description:

The definitized System Specification defines the system capabilities that satisfy the customer/user needs, and will be included in the productized system.

General Content:

- System Capabilities
- System Interfaces
- Performance
- Rationale

Development:

The definitized System Specification is a major output of the System Architecture phase.

External/Internal:

The definitized System Specification is an external document.

Related DoD-STD-2167A Sections:

The table below shows a mapping of DoD-STD-2167A sections to the above general content, for this CDRL item. Following the table is an indented list of the DoD-STD-2167A sections/subsections included in the mapping.

2167A Section	System Capabilities	System Interfaces	Performance	Rationale
System/Segment Specification				
3. System requirements				
3.1 Definition	X	X		
3.2 Characteristics	X	X	X	
3.3 Design and Construction	X		X	
3.4 Documentation	X		X	
3.5 Logistics	X		X	
3.6 Personnel and training	X		X	
3.7 Characteristics of subordinate elements	X			
3.8 Precedence	X			
3.9 Qualification	X		X	
3.10 Standard Sample	X		X	
3.11 Preproduction sample, periodic production sample, pilot, or	X		X	
4. Quality assurance provisions	X		X	

The 2167A section/subsections included in the above mapping are listed below:

System/Segment Specification

3. System requirements
   - 3.1 Definition
   - 3.2 Characteristics
   - 3.3 Design and Construction
   - 3.4 Documentation
   - 3.5 Logistics
   - 3.6 Personnel and training

3.7 Characteristics of subordinate elements
3.8 Precedence
3.9 Qualification
3.10 Standard Sample
3.11 Preproduction sample, periodic production sample, pilot, or pilot lot
4. Quality assurance provisions
4.1 Responsibility for inspection
4.2 Special tests and examinations
4.3 Requirements cross reference

7. SOFTWARE DESIGN

Type: Document.

Description:

The definitized Software Design details the implementation of the operational system software.

General Content:

- Software Architecture
- Interfaces
- Data
- Performance
- Rationale

Development:

The Software Design is definitized during the Productization and Production phase.

External/Internal:

The definitized Software Design is an external document.

Related DoD-STD-2167A Sections:

The table below shows a mapping of DoD-STD-2167A sections to the above general content, for this CDRL item. Following the table is an indented list of the DoD-STD-2167A sections/subsections included in the mapping.

2167A Section	Software Architecture	Interfaces	Data	Performance	Rationale
Software Product Specification					
3. Requirements					
3.1 Software design (Software Design Document)	X	X	X	X	
3.2 CSCI (component) source code listings	X	X	X		
3.3 Compiler/assembler	X				
3.4 Measured resource utilization				X	

The 2167A section/subsections included in the above mapping are listed below:

Software Product Specification

3. Requirements
   3.1 Software Design (includes or references Software Design Document(s), as defined under Component Design)
   3.2 CSCI (component) source code listings (includes or references source code listings)
   3.3 Compiler/assembler
   3.4 Measured resource utilization

8. COMPONENT SPECIFICATION

Type: Document.

Description:

The definitized Component Specification details the capabilities of a Component.

General Content:

- Capabilities
- Interfaces
- Performance
- Qualification
- Rationale

Development:

There is a definitized Component Specification for each Component developed in the Software Growing phase.

External/Internal:

The definitized Component Specification is an external work product.

Related DoD-STD-2167A Sections:

The table below shows a mapping of DoD-STD-2167A sections to the above general content, for this CDRL item. Following the table is an indented list of the DoD-STD-2167A sections/subsections included in the mapping.

2167A Section	Capabilities	Interfaces	Performance	Qualification	Rationale
Software Requirements Specification					
3. Engineering requirements					
3.1 CSCI external interface requirements		X			
3.2 CSCI capability requirements	X				
3.3 CSCI internal interfaces		X			
3.4 CSCI data element requirements		X			
3.5 Adaptation requirements		X			
3.6 Sizing and timing requirements			X		
3.7 Safety requirements	X				
3.8 Security requirements	X				
3.9 Design constraints	X				
3.10 Software quality factors				X	
3.11 Human performance/human engineering requirements	X				
3.12 Requirements traceability	X				
4. Qualification requirements				X	
Interface Requirements Specification					
3. Interface Specification		X			

The 2167A section/subsections included in the above mapping are listed below:

- Software Requirements Specification
  - 3. Engineering requirements
    - 3.1 CSCI external interface requirements
    - 3.2 CSCI capability requirements
    - 3.3 CSCI internal interfaces
    - 3.4 CSCI data element requirements
    - 3.5 Adaptation requirements
    - 3.6 Sizing and timing requirements
    - 3.7 Safety requirements
    - 3.8 Security requirements
    - 3.9 Design constraints
    - 3.10 Software quality factors
    - 3.11 Human performance/human engineering requirements
    - 3.12 Requirements traceability
  - 4. Qualification requirements
  - 4.1 Qualification methods

4.2 Special qualification requirements
- Interface Requirements Specification
  3. Interface Specification
     3.1 Interface diagrams
     3.X (Interface name and project unique identifier)

9. COMPONENT DESIGN

Type: Document.

Description:

The definitized Component Design details the implementation of a Component.

General Content:

- Architecture
- Design
- Data
- Interfaces
- Rationale

Development:

There is a definitized Component Design document for each Component developed in the Software Growing phase.

External/Internal:

The definitized Component Design is an external work product.

Related DoD-STD-2167A Sections:

The table below shows a mapping of DoD-STD-2167A sections to the above general content, for this CDRL item. Following the table is an indented list of the DoD-STD-2167A sections/subsections included in the mapping.

2167A Section	Architec-ture	Design	Data	Interfaces	Rationale
Software Design Docu-ment					
3. Preliminary Design	X				
4. Detailed Design		X			
5. CSCI (component) data			X		
6. CSCI (component) data files			X		
7. Requirements traceability		X			
Interface Design Docu-ment					
3. Interface Design				X	

The 2167A section/subsections included in the above mapping are listed below:

- Software Design Document

3. Preliminary Design
    3.1 CSCI (component) overview
    3.2 CSCI (component) design description
4. Detailed Design
    4.X (CSC (component) name and project unique identifier)
5. CSCI (component) data
6. CSCI (component) data files
    6.1 Data file to CSC/CSU (component) cross reference
    6.X (Data file name and project unique identifier)
    7. Requirements traceability

- Interface Design Document
    3. Interface Design
        3.1 Interface diagrams
        3.X (Interface name and project unique identifier)

Acronyms

Acronym	*Meaning*
CDRL	Contract Data Requirements List
DoD	(United States) Department of Defense
IBM	International Business Machines
SFLC	Software First Life Cycle
SOW	Statement of Work
STARS	Software Technology for Adaptable, Reliable Systems

References

(Boeh88) Boehm, Barry W., "A Spiral Model of Software Development and Enhancement", IEEE Computer, May 1988.

(DOD2167A) Department of Defense, DoD-STD-2167A, Military Standard, *Defense System Software Development*, 1988

(DODHDBK) Department of Defense, MIL-HDBK-287, Military Handbook, *A Tailoring Guide for DoD-STD-2167A, Defense System Software Development*, 1989

(Gree89) Greene, Joseph S. Jr., Colonel, USAF, "Position Paper - Subject: Software-First", Defense Advanced Research Projects Agency, 1989.

(IBM1240) IBM System Integration Division, *Software-First Life Cycle - Final Definition*, CRDL Sequence No. 01240, January 1990

(IBM1840) IBM System Integration Division, *Proposed Software-First Life Cycle CDRLs - Preliminary*, CRDL Sequence No. 01840, March 1990

(IBM1850) IBM System Integration Division, *SFLC DIDs with 2167A Sections - Preliminary* CRDL Sequence No. 01850, April 1990

(IBM1860) IBM System Integration Division, *Proposed Software-First Life Cycle CDRLs - Final*, CRDL Sequence No. 01860, May 1990

(IBM1870) IBM System Integration Division, *Proposed Software-First Life Cycle DIDs - Final*, CRDL Sequence No. 01870, June 1990

(JLC89) Department of Defense, Joint Logistics Commanders, JPCG-CRM, CSM, *Software Development Under DoD-STD-2167A: An Examination of Ten Key Issues*, 1989

(STAR87) STARS Joint Program Office, Greene, Joseph S. Jr., Colonel, "Software-First Systems Development Standard For Systems-In-The-Large", *Attachment A.5, STARS SOW Materials*, 1987.

Appendix A. Software Development Plan

This appendix describes how the DoD-STD-2167A CDRL item DID (Data Item Description), for the Software Development Plan (SDP) can be tailored for use as the SFLC Software Development Plan.

Differences Between DoD-STD-2167A and SFLC:

The Software Development Plan (SDP), as defined in SDP DID, "decribes a contractor's plan for conducting software development." Also, as stated in the SDP DID, "The SDP is used to provide the Government insight into the organization(s) responsible for performing software development and the methods and procedures to be followed by these organization(s)." The SDP is intended to provide the "roadmap for the software development and is a living document that may be changed as events occur (JLC89)."

In the SFLC, there are two levels of planning, system-level and component level. Thus, there will be a single system-level Software Development Plan (SDP) and multiple component-level SDPs. All of them will be evolving documents. In the Preliminary System Analysis phase, a preliminary version of the system-level SDP is produced. The preliminary version should contain sufficient information to provide direction to development of system prototypes and the productized system. This includes an overall incremental build plan for the system and a prototyping plan for the initial system prototype. In the System Architecture phase, the system-level SDP will expand as the system prototypes evolve into a full-capability prototype. In the Productization and Production phase, where the full-capability prototype is converted into a productized system, the system-level SDP will expand to include the system testing required of the productized system. In the Software Growing phase, each component-level SDP will expand as the associated component prototype evolves into a full-capability prototype, and then into a productized component.

General Tailoring Guidelines:

SDPs will provide planning and documentation of the management, methods, and procedures to be used in developing the software for the system and component protoypes, as well as the productized system and its associated components. As mentioned above, all of the SDPs will be evolving documents. They will contain the information defined in the DoD-STD-2167A SDP DID, tailored to the unique requirements of the SFLC phases and the associated technologies, such as rapid prototyping and reusability. For example, multiple management plans are needed, including incremental build, prototyping, and productization plans. Also, in addition to describing formal qualification testing and software product evaluation, the SDP should describe the prototype test and evaluation approaches. A software measurement/metrics function should also be included in the SDP.

Specific Tailoring Guidelines:

The DID text for each section is provided below, followed by specific tailoring guidelines for that section, as appropiate. The tailoring described is focused on what is required to allow the SDP DID to be used for the SFLC Software Development Plan. Additional tailoring of each section will be needed, to make it reflect the unique requirements of the system and the specific software development methodologies being used.

DoD-STD-2167A DID: Software Development Plan

IDENTIFICATION NUMBER: DI-MCCR-80030A

3. Software development management

This section shall be numbered 3 and shall be divided into the following paragraphs and subparagraphs to describe the planning associated with software development management activities.

3.1 Project organization and resources

This paragraph shall be numbered 3.1 and shall be divided into the following subparagraphs to describe the project organization and the project resources of the contractor.

3.1.1 Contractor facilities

This subparagraph shall be numbered 3.1.1 and shall provide a description of the contractor's facilities to be used for the contracted effort. This subparagraph shall highlight secure areas and briefly identify the nature of the secure activity. This subparagraph shall also highlight the location of project specific resources such as the software engineering environment and software test environment.

Tailoring Guidelines:

A description of the the contractor's facilities should be included in the preliminary SDP and expanded as the prototypes evolve into a full-capability prototype, and then into a productized system (or component). Where used, standard tools and interfaces should be identified.

3.1.2 Government furnished equipment, software, and services

This subparagraph shall be numbered 3.1.2 and shall summarize all Government furnished equipment, software, services, and facilities required for the contracted effort. A schedule detailing when these items will be needed shall also be included. 'This subparagraph shall highlight all required items not listed in the System/Segment Specification, Prime Item Development Specification, or Critical Item Development Specification, as applicable.

Tailoring Guidelines:

A description of the Government furnished equipment, software, and services should be included in the preliminary SDP and expanded as the prototypes evolve into a full-capability prototype, and then into a productized system (or component). Where used, standard tools and interfaces should be identified.

3.1.3 Organizational structure

This subparagraph shall be numbered 3.1.3 and shall provide an overview of the contractor's software project organizational structure. This subparagraph shall identify the authority and responsibilities of each organization. This information may be provided graphically.

Tailoring Guidelines:

None

3.1.4 Personnel

This subparagraph shall be numbered 3.1.4 and shall identify the total number of personnel necessary to complete the software development project. This summary shall indicate the total number

of personnel for project management, software engineering, formal software testing, software product evaluations, software configuration management and any other functions identified in this plan.

Tailoring Guidelines:

None

3.2 Schedule and milestones

This paragraph shall be numbered 3.2 and shall be divided into the following subparagraphs.

3.2.1 Activities

This subparagraph shall be numbered 3.2.1 and shall briefly describe each software development activity of the project and its associated schedule, based on the contract master schedule (if applicable). The development schedule shall also indicate all significant events, such as reviews, audits, key meetings, etc. The schedule may be provided graphically. For each activity, the schedule shall indicate:

a. Activity initiation

b. Availability of draft and final copies of formal and informal documentation

c. Activity completion

d. Areas of high risk.

Tailoring Guidelines:

This subparagraph should include the multiple activities and plans required to evolve the prototypes into a full-capability prototype, and then to convert the full-capability prototype into a productized system (or component). This includes an overall incremental build plan for the system or component, a prototyping plan for the initial prototype, and a productization plan for converting the full-capability prototype into a productized system or component.

3.2.2 Activity network

This subparagraph shall be numbered 3.2.2 and shall describe the sequential relationship among the activities of the project. This subparagraph shall include identification of those activities that impose the greatest time restrictions on project completion and those activities with an excess of time for completion. This information may be provided graphically.

Tailoring Guidelines:

None

3.2.3 Source identification

This subparagraph shall be numbered 3.2.3 and shall identify and describe the source of the required resources (software, firmware, and hardware) for the software development effort. This subparagraph shall provide a plan for obtaining the required resources and shall indicate the need date and availability of each resource item.

Tailoring Guidelines:

None

3.3 Risk management

This paragraph shall be numbered 3.3 and shall describe the contractor's procedures for managing areas of risk to successful project completion. This paragraph shall:

a. Identify the areas of risk to successful project completion and prioritize them.

b. Identify the constituent risk factors that contribute to the potential occurrence of each risk.

c. Document procedures for monitoring the risk factors and for reducing the potential occurrence of each risk.

d. Identify contingency procedures for each area of risk, as appropriate.

Tailoring Guidelines:

None

3.4 Security

This paragraph shall be numbered 3.4 and shall describe the contractor's plans for implementing the security requirements of the contract.

Tailoring Guidelines:

None

3.5 Interface with associate contractors

This paragraph shall be numbered 3.5 and shall describe the contractor's plan for coordinating design and data management efforts to ensure compatibility at interfaces with associate contractors (i.e. where two or more contractors are participating in development or production of the system).

Tailoring Guidelines:

None

3.6 Interface with software IV and V agent(s)

This paragraph shall be numbered 3.6 and shall describe the contractor's plans for interfacing with the software independent verification and validation (IV and V) agent(s), if applicable.

Tailoring Guidelines:

None

3.7 Subcontractor management

This paragraph shall be numbered 3.7 and shall describe the contractor's plans for managing subcontractors.

Tailoring Guidelines:

None

3.8 Formal reviews

This paragraph shall be numbered 3.8 and shall describe the contractor's internal procedures for preparing for and conducting formal reviews.

Tailoring Guidelines:

Internal procedures for preparing for and conducting prototype demonstrations and evaluations should also be included in this paragraph.

3.9 Software development library

This paragraph shall be numbered 3.9 and shall describe the software development library (SDL) to be used by the contractor for controlling the software and associated documentation. This paragraph shall include a description of the contractor's procedures and methods for establishing and implementing the SDL and the contractor's access and control procedures for data stored in the SDL.

Tailoring Guidelines:

None

3.10 Corrective action process

This paragraph shall be numbered 3.10 and shall describe the corrective action process to be implemented.

Tailoring Guidelines:

None

3.11 Problem/change report

This paragraph shall be numbered 3.11 and shall describe the format to be used for problem/change reports. These reports are used to document problems detected in the software or its documentation and to describe the corrective action needed to resolve the problems. Candidate data items for the report include:

a. System or project name - The name of the system or development project to which this report applies.

b. Originator - The name, telephone number, and designation of the persons or organization(s) submitting the report.

c. Problem number - The assigned problem number.

d. Problem name- A brief phrase descriptive of the problem and descriptive of similar problems, if applicable.

e. Software element or document affected - The specific software element(s), document(s) paragraphs(s), or both to which the report applies, including appropriate configuration identification and version number, if applicable.

f. Origination date - The date the report is first submitted.

g. Category and priority - See Appendix C of DOD-STD-2167A.

h. Description of problem - A description of the problem and the conditions, inputs, and equipment configuration under which the problem arises. A description of the activities leading up to problem occurrence. Sufficient problem information to permit duplication and analysis. Relationship to other reported problems and modifications.

i. Analyst - The name, telephone number, and organization of the individual assigned to analyze the problem.

j. Date assigned - The date the analyst was assigned.

k. Date complete - The date the analysis was completed.

l. Analysis time - The time required to analyze the problem.

m. Recommended solution - After analysis of the problem, the recommended solution and alternative solutions, if available. The nature of the recommended solution by a short descriptive phrase. When applicable, supporting rationale and test results.

n. Impacts - The cost, schedule, and interface impacts if the solution is approved. Also, performance impacts if the solution is not approved. As applicable, the impact on other systems, configuration items, other contractors, system employment, integrated logistics support, system resources, training, etc.

o. Problem status - The problem status designated by configuration control procedures.

p. Approval of solution - To be designated by the cognizant configuration control authority.

q. Follow-up action - Actions following resolution of the problem.

r. Corrector - The name, telephone number, and organization of the individual correcting the problem.

s. Correction date - The date the problem was corrected.

t. Version number - The version in which the problem was corrected.

u. Correction time- The time required to correct the problem.

v. Implementation solution - A brief description of the implemented solution to the problem.

Tailoring Guidelines:

None

4. Software engineering

This section shall be numbered 4 and shall be divided into the following paragraphs and subparagraphs to describe the planning associated with software engineering activities.

4.1 Organization and resources - software engineering

This paragraph shall be numbered 4.1 and shall be divided into the following subparagraphs to describe the organization(s) responsible and the resources necessary for software engineering activities.

Tailoring Guidelines:

None

4.1.1 Organizational structure - software engineering

This subparagraph shall be numbered 4.1.1 and shall describe the organization(s) responsible for performing the software engineering activities. This subparagraph shall include the authority and responsibilities of each organization and its relationship to other organizational entities such as the organization(s) responsible for performing software quality evaluations. If more than one organization is involved, the precise structure, personnel, and resources of each organization and their interrelationships shall be highlighted.

Tailoring Guidelines:

None

4.1.2 Personnel - software engineering

This subparagraph shall be numbered 4.1.2 and shall describe the number and skill levels of personnel who will perform the software engineering activities. The personnel shall be described by title and minimum qualifications for the position. In addition, this subparagraph shall specify any requirements unique to particular positions, such as geographic location, security level, extended hours, etc.

Tailoring Guidelines:

None

4.1.3 Software engineering environment

This subparagraph shall be numbered 4.1.3 and shall be divided into the following subparagraphs to identify and describe the plans for establishing and maintaining the resources (software, firmware, and hardware) necessary to perform the software engineering activities.

4.1.3.1 Software items: This subparagraph shall be numbered 4.1.3.1 and shall identify the software items, such as operating systems, compilers, code auditors, dynamic path analyzers, test drivers, preprocessors, test data generators, post-processors, etc., necessary to perform the software engineering activities. This subparagraph shall describe the purpose of each item and shall identify any classified processing or security issues associated with the software items.

Tailoring Guidelines:

None

4.1.3.2 Hardware and firmware items: This subparagraph shall be numbered 4.1.3.2 and shall identify the computer hardware, interfacing equipment, and firmware items that will be used in the software engineering environment. This subparagraph shall describe the purpose of each item and shall identify any classified processing or security issues associated with the hardware or firmware items.

Tailoring Guidelines:

None

4.1.3.3 Proprietary nature and government rights: This subparagraph shall be numbered 4.1.3.3 and shall identify the proprietary nature and Government rights associated with each item of the software engineering environment.

Tailoring Guidelines:

None

4.1.3.4 Installation, control, and maintenance: This subparagraph shall be numbered 4.1.3.4 and shall identify the contractor's plans for installing and testing each item of the software engineering environment prior to its use. This subparagraph shall also describe the contractor's plans for controlling and maintaining each item of the software engineering environment.

Tailoring Guidelines:

None

4.2 Software standards and procedures

This paragraph shall be numbered 4.2 and shall be divided into the following subparagraphs to describe the software standards and procedures the contractor plans to use.

4.2.1 Software development techniques and methodologies

This subparagraph shall be numbered 4.2.1 and shall identify and describe the techniques and methodologies the contractor plans to use to perform:

a. Software Requirements Analysis

b. Preliminary Design

c. Detailed Design

d. Coding and CSU Testing

e. CSC Integration and Testing

f. CSCI Testing.

Tailoring Guidelines:

The SFLC, and the associated techniques and methodologies that are used, should be described or referenced in this subparagraph.

4.2.2 Software development files

This subparagraph shall be numbered 4.2.2 and shall define the contractor's plans, including the responsible organization(s), for the creation and maintenance of software development files (SDFs). This subparagraph shall define the format and contents of the SDFs and describe the procedures for maintaining SDFs.

Tailoring Guidelines:

None

4.2.3 Design standards

This subparagraph shall be numbered 4.2.3 and shall describe the design standards the contractor plans to use in developing the software.

Tailoring Guidelines:

None

4.2.4 Coding standards

This subparagraph shall be numbered 4.2.4 and shall describe the coding standards the contractor plans to use in developing the software.

Tailoring Guidelines:

None

4.3 Non-developmental software

This paragraph shall be numbered 4.3 and shall identify and describe each non-developmental software item, such as commercially available, reusable, and Government furnished software, to be incorporated into the deliverable software. This subparagraph shall briefly describe the rationale for the use of each non-developmental software item.

Tailoring Guidelines:

None

5. *Formal qualification testing*

This section shall be numbered 5 and shall be divided into the following paragraphs and subparagraphs to describe the planning associated with formal qualification testing activities.

5.1 Organization and resources - formal qualification testing

This paragraph shall be numbered 5.1 and shall be divided into the following subparagraphs to describe the organization(s) responsible and the resources necessary for formal qualification testing.

5.1.1 Organizational structure - formal qualification testing

This subparagraph shall be numbered 5.1.1 and shall describe the organization(s) responsible for performing formal qualification testing. The subparagraph shall include the authority and responsibilities of each organization and its relationship to other organizational entities such as the organization(s) responsible for performing software engineering. If more than one organization is involved, the precise structure, personnel, and resources of each organization and their interrelationships shall be highlighted.

Tailoring Guidelines:

None

5.1.2 Personnel - formal qualification testing

This subparagraph shall be numbered 5.1.2 and shall describe the number and skill levels of personnel who will perform the formal qualification testing activities. The personnel shall be described by title and minimum qualifications for the position. In addition, this subparagraph shall specify any requirements unique to particular positions, such as geographic location, security level, extended hours, etc.

Tailoring Guidelines:

None

5.2 Test approach/philosophy

This paragraph shall be numbered 5.2 and shall describe the contractor's approach/philosophy for performing formal qualification testing.

Tailoring Guidelines:

This paragraph should describe the various test approaches/philosophies that will be used to test and evaluate the productized system or productized components, as well as the associated prototypes.

5.3 Test planning assumptions and constraints

This paragraph shall be numbered 5.3 and shall describe any assumptions that were made in test planning and any constraints imposed upon formal qualification testing by the contracting agency.

Tailoring Guidelines:

This paragraph should describe the different SFLC test assumptions/philosophies that will be used in testing and evaluating the productized system or productized components, as well as the associated prototypes.

6. *Software product evaluations*

This section shall be numbered 6 and shall be divided into the following paragraphs and subparagraphs to describe the planning associated with software product evaluation activities.

6.1 Organization and resources - software product evaluations

This paragraph shall be numbered 6.1 and shall be divided into the following subparagraphs to describe the organization(s) responsible and the resources necessary for software product evaluations.

6.1.1 Organizational structure - software product evaluations

This subparagraph shall be numbered 6.1.1 and shall describe the organization(s) responsible for performing the software product evaluations. This subparagraph shall include the authority and responsibilities of each organization and its relationship to other organizational entities such as the organization(s) responsible for performing software engineering. If more than one organization is involved, the precise structure, personnel, and resources of each organization and their interrelationships shall be highlighted.

Tailoring Guidelines:

None

6.1.2 Personnel - software product evaluations

This subparagraph shall be numbered 6.1.2 and shall describe the number and skill levels of personnel who will perform software product evaluations. The personnel shall be described by title and minimum qualifications for the position. In addition, this subparagraph shall specify any requirements unique to particular positions, such as geographic location, security level, extended hours, etc.

Tailoring Guidelines:

None

6.2 Software product evaluations procedures and tools

This paragraph shall be numbered 6.2 and shall be divided into the following subparagraphs.

6.2.1 Procedures

This subparagraph shall be numbered 6.2.1 and shall identify and describe the procedures that will be used to evaluate the software and associated documentation.

Tailoring Guidelines:

None

6.2.2 Tools

This subparagraph shall be numbered 6.2.2 and shall identify and describe the tools to be used in the software product evaluations. Tool descriptions shall identify each tools's purpose in the evaluation process. To reduce duplication, references any be made to tools that are also used in the software engineering or software test environments.

Tailoring Guidelines:

None

6.3 Subcontractor products

This paragraph shall be numbered 6.3 and shall describe the contractor's plans and procedures for evaluating the adequacy of requirements established for subcontractors and for evaluating subcontractor products.

Tailoring Guidelines:

None

6.4 Software product evaluation records

This paragraph shall be numbered 6.4 and shall describe the contractor's plans for preparing and maintaining records of each product evaluation performed. It shall identify the formats to be used and the information to be recorded for each evaluation. It shall also describe plans for maintaining the records and for making them available for contracting agency review.

Tailoring Guidelines:

None

6.5 Activity-dependent product evaluations

This paragraph shall be numbered 6.5 and shall be divided into subparagraphs to describe the contractor's plans for conducting product evaluations of each software development product. This paragraph shall explain any planned modifications or additions to the evaluation criteria required by DOD-STD-2167A. The following subparagraphs shall address plans for product evaluations conducted during each of the software development activities (i.e., System Requirements Analysis/Design, Software Requirements Analysis, Preliminary Design, Detailed Design, Coding and CSU Testing, CSC Integration and Testing, CSCI Testing, and System Integration and Testing).

Tailoring Guidelines:

This paragraph should describe the different sets of evaluation criteria required to evaluate, for each of the SFLC phases, the software development work products associated with the productized system or productized components and their corresponding prototypes.

6.5.X Software products evaluation - (activity name)

This subparagraph shall be numbered 6.5.X (beginning with 6.5.1) and shall describe the contractor's plans for conducting evaluations of each of the products of an activity. The description shall identify the specific products to be evaluated. For each product to be evaluated, the evaluation criteria to be used and the evaluation procedures and tools to be employed shall be identified. For evaluations performed on items contained in SDFs, the method of selecting the sample and the percentage of the items to be evaluated shall be specified.

Tailoring Guidelines:

None

7. Software configuration management

This section shall be numbered 7 and shall be divided into the following paragraphs and subparagraphs to describe the planning associated with software configuration management (CM) activities.

7.1 Organization and resources - configuration management

This paragraph shall be numbered 7.1 and shall be divided into the following subparagraphs to describe the organization(s) responsible and the resources necessary for configuration management.

7.1.1 Organizational structure - configuration management

This subparagraph shall be numbered 7.1.1 and shall describe the organization(s) responsible for performing configuration management. This subparagraph shall include the authority and responsibilities of each organization and its relationship to other organizational entities such as the organization(s) responsible for performing software engineering. If more than one organization is involved, the precise structure, personnel, and resources of each organization and their interrelationships shall be highlighted.

Tailoring Guidelines:

None

7.1.2 Personnel - configuration management

This subparagraph shall be numbered 7.1.2 and shall describe the number and skill levels of personnel who will perform configuration management. The personnel shall be described by title and minimum qualifications for the position. In addition, this subparagraph shall specify any requirements unique to particular positions, such as geographic location, security level, extended hours, etc.

Tailoring Guidelines:

None

7.2 Configuration identification

This paragraph shall be numbered 7.2 and shall be divided into the following subparagraphs.

7.2.1 Development configuration identification

This subparagraph shall be numbered 7.2.1 and shall identify the contractor's internal Developmental Configuration(s) to be used in the development of the CSC(s). For each Developmental Configuration identified, the method of establishing it shall be described and the contents shall be listed.

Tailoring Guidelines:

None

7.2.2 Identification methods

This subparagraph shall be numbered 7.2.2 and shall describe the methods to be used in identifying (e.g., naming, marking, numbering) CSCI(s), CSCs, CSUs, and documentation. This subparagraph shall also describe how revisions to the CSCIs, CSCs, CSUs, and documentation shall be identified.

Tailoring Guidelines:

None

7.3 Configuration control

This paragraph shall numbered 7.3 and shall be divided into the following subparagraphs to provide a detailed description of the procedures to be used in controlling changes to and maintaining the Developmental Configuration(s) and internally controlled documentation.

7.3.1 Flow of configuration control

This subparagraph shall be numbered 7.3.1 and shall describe the process by which problems and changes are submitted, reviewed and subsequently approved or disapproved. This description may be accomplished graphically by a configuration control flow chart (see Figure 1).

Tailoring Guidelines:

None

7.3.2 Reporting documentation

This subparagraph shall be numbered 7.3.2 and shall be divided into the following subparagraphs to describe or reference the description of the reporting documentation, such as Specification Change Notices and Engineering Change Proposals to be used in controlling software problems and changes.

7.3.2.X (Report name): This subparagraph shall be numbered 7.3.2.X beginning with 7.3.2.1) and shall describe or reference the format, contents, and instructions for completing the report.

Tailoring Guidelines:

None

7.3.3 Review procedures

This subparagraph shall be numbered 7.3.3 and shall be divided into the following subparagraphs to describe the purpose of, and the procedures to be employed by, any review boards associated with the flow of configuration control.

7.3.3.X (Review board name) procedures: This subparagraph shall be numbered 7.3.3.X (beginning with 7.3.3.1) and shall describe the purpose of and the procedures to be followed by the review board. This subparagraph shall also describe how the procedures set by the review board, in conjunction with the configuration identification scheme, provide historical traceability.

Tailoring Guidelines:

None

7.3.4 Storage, handling, and delivery of project media

This subparagraph shall be numbered 7.3.4 and shall describe the methods and procedures to be used to formally corporate storage, handling, and delivery of deliverable software and documentation (including master copies during the development process).

Tailoring Guidelines:

None

7.3.5 Additional control

This subparagraph shall be numbered 7.3.5 and shall identify any additional configuration control activities not discussed above.

Tailoring Guidelines:

None

7.4 Configuration status accounting

This paragraph shall be numbered 7.4 and shall define the configuration status accounting system to be used. The content format and purpose of the status accounting records and reports shall be described.

Tailoring Guidelines:

None

7.5 Configuration audits

This paragraph shall be numbered 7.5 and shall describe the contractor's plans for supporting or conducting configuration audits, as applicable. The description of how the configuration status accounting reports and records should be used in conducting these audits shall be included.

Tailoring Guidelines:

None

7.6 Preparation for specification authentication

This paragraph shall be numbered 7.6 and shall describe the contractor's procedures to prepare for and respond to authentication of the applicable specifications. This paragraph shall include the procedures for:

a. Submitting specifications to the contracting agency for review and authentication.

b. Ensuring the incorporation of approved changes.

c. Updating the configuration status accounting reports to reflect approved baseline(s).

Tailoring Guidelines:

None

7.7 Configuration management major milestones

This paragraph shall be numbered 7.7 and shall identify the major internal and Government milestones related to software configuration management for the contractual effort.

Tailoring Guidelines:

None

8. Other software development functions

This section shall be numbered 8 and shall be divided into the following paragraphs and subparagraphs to describe any other contractor functions involved in the software development effort.

Tailoring Guidelines:

It is recommended that a software metrics function be established and described in this section. This should include what information will be collected, and how it will be used to measure and evaluate the productivity and quality of the software development effort.

8.X (Function name)

This paragraph shall be numbered 8.X (beginning with 8.1) and shall describe a function to be performed. This paragraph shall be divided into the following subparagraphs to describe the organizational structure, resources, and the methods and procedures necessary to perform the function.

8.X.1 Organizational structure (function name)

This subparagraph shall be numbered 8.X.1 (beginning with 8.1.1) and shall describe the organization(s) responsible for performing the function. This subparagraph shall include the authority and responsibilities of each organization and its relationship to other organizational entities such as the organization(s) responsible for performing configuration management. If more than one organization is involved, the precise structure, personnel, and resources of each organization and their interrelationships shall be highlighted.

8.X.2 Personnel - (function name)

This subparagraph shall be numbered 8.X.2 and shall describe the number and skill levels of personnel who will perform the function. The personnel shall be described by title and minimum qualifications for the position. In addition, this subparagraph shall specify any requirements unique to particular positions, such as geographic location, security level, extended hours, etc.

8.X.3 Other resources - (function name)

This subparagraph shall be numbered 8.X.3 and shall identify and describe any other resources necessary for performing the function. For each resource, this subparagraph shall briefly describe the aspect of the function that requires the resource.

8.X.4 Methods and procedures - (function name)

This subparagraph shall be numbered 8.X.4 and shall describe the methods and procedures to be used to perform the function.

Appendix B. System Description

This appendix describes how the DoD-STD-2167A CDRL item DID (Data Item Description), for the System/Segment Design Document (SSDD), can be tailored for use as the SFLC System Description (SD).

Differences Between DoD-STD-2167A and SFLC:

The System/Segment Design Document (SSDD), as defined in SSDD DID, "describes the design of a system/segment and its operational and support environment. It describes the organization of a system or segment as composed of Hardware Configuration Items (HWCIs), Computer Software Configuration Items (CSCIs), and manual operations." The SSDD, on a typical DoD-STD-2167A project, is produced as a companion design document to the System/Segment Specification.

In the SFLC, a preliminary version of the System Description is produced during the Preliminary System Analysis phase. The preliminary version includes a definitized description of the system mission and operational need and a preliminary description of the system architecture. A fully definitized System Description is written, at the end of the System Architecture phase, after the full-capability system prototype is completed. At this point the reusable components have been integrated with the new components of the system, and the system architecture for the productized system (which will be developed from the the full-capability system prototype) can be defined in very definitive terms.

General Tailoring Guidelines:

It is recommended that CSCIs, as identified in the SSDD, be defined as either closely related sets of software components and/or individual high level software components. The choices will, of course, depend on the specific application. This will allow the System Description to define the system architecture, relationships between components, external/internal interfaces, operational scenarios, etc., using the SSDD DID. Where used, standard interfaces should also be identified and described. In addition, system-level diagrams should provide a nomenclature for distinguishing between those components which will come from the reusable component repository and those which will be newly developed.

The level of detail contained in the preliminary version of the System Description will be project dependent. In many cases, only preliminary definitions of the system architecture, external/internal interfaces to hardware and software components, etc., can be included, since much of the information will not be available until a full-capability system prototype has been developed and a definitized System/Segment Specification written. At that time, the system capabilities and requirements, and the target hardware configuration, will be fully defined. However, sufficient detail should be present in the preliminary version to provide guidance to the specification, design, and implementation of the initial system prototype(s).

The definitized System Description will contain the finalized definitions of the system architecture, external/internal interfaces to hardware and software components, etc. In addition, it will include a formal allocation of System/Segment Specification requirements to system components.

For those projects using an object-oriented design (OOD) methodology, components should be high level objects or related groups of objects. Component interfaces, identified in the SSDD DID, would then define the flow of messages (operations) between components.

Specific Tailoring Guidelines:

The SD will evolve from a preliminary version, produced during the Preliminary System Analysis phase, to a definitized version, produced at the end of the System Architecture phase, after the full-capability system prototype is completed. To provide guidance to this evolution, the DID text for each section of the SSDD (beginning with section 3.) is provided below, followed by specific tailoring guidelines for that section, as appropiate. The tailoring described is focused primarily on what is required to allow the SSDD DID to be used for the SFLC System Description document as it evolves. Additional tailoring of each section will be needed, depending on the unique requirements of the system and the specific software development methodologies being used.

DoD-STD-2167A DID: System/Segment Design Document

IDENTIFICATION NUMBER: DI-CMAN-80534

Note: The word "system" used in the DID text that follows can mean either a system or a segment, as applicable.

3. Operational concepts

This section shall be numbered 3 and shall be divided into the following paragraphs and subparagraphs to describe the operational concepts of the system.

3.1 Mission

This subparagraph shall be numbered 3.1 and shall be divided into the following subparagraphs.

3.1.1 User Needs

This subparagraph shall be numbered 3.1.1, shall summarize the user needs that are to be met by the system and shall reference the document(s) in which these needs are stated.

Tailoring Guidelines:

The user needs should be definitized during the Preliminary System Analysis phase and be included in the preliminary System Description, which is a major workproduct of that phase. The user need definition should include:

- Identification and evaluation of user capabilities of existing system
- Identification of additional user capabilities required in the new system
- Prioritization of capabilities
- Identification of constraints

3.1.2 Primary mission(s)

This subparagraph shall be numbered 3.1.2 and shall describe the primary mission(s) of the system.

Tailoring Guidelines:

The primary mission(s) should be definitized during the Preliminary System Analysis phase and be included in the preliminary System Description, which is a major workproduct of that phase.

3.1.3 Secondary mission(s)

This subparagraph shall be numbered 3.1.3 and shall describe the secondary mission(s) of the system.

Tailoring Guidelines:

Same as primary mission(s) subparagraph.

3.2 Operational environment

This paragraph shall be numbered 3.2 and shall describe the environment in which the system is intended to be employed.

Tailoring Guidelines:

A preliminary description of the operational environment should be produced during the Preliminary System Analysis phase and expanded as the system prototypes evolve into a full-capability prototype. A final description of the operational environment will be written after the full-capability system prototype is completed and will be included in the definitized System Description, which is a major workproduct of the System Architecture phase.

3.3 Support environment

This paragraph shall be numbered 3.3 and shall describe the support environment for the operational system during the Production and Deployment phase or the system life cycle.

Tailoring Guidelines:

The support environment will be finalized after the full-capability system prototype is completed, and will be described in the definitized System Description. The need for early preliminary definitions of various aspects of the support environment as defined in the subparagraphs below, will be project dependent. If needed, preliminary definitions required to support the development and operation of system prototypes should be produced during the Preliminary System Analysis phase and expanded, as necessary, as the system prototypes evolve into a full-capability prototype.

3.3.1 Support concept

This subparagraph shall be numbered 3.3.1 and shall describe the support concept for the system. This subparagraph shall include the following:

a. Use of multipurpose or automated test equipment

b. Repair versus replacement criteria

c. Levels of maintenance

d. Maintenance and repair cycles

e. Government and contractor support

f. Accessibility

g. Other.

Tailoring Guidelines:

See tailoring guidelines for paragraph 3.3.

3.3.2 Support facilities

This subparagraph shall be numbered 3.3.2 and shall describe the system support facilities and equipment to be used during the Production and Deployment phase of the system life cycle. A quantitative description of existing facilities and equipment shall be provided in sufficient detail so that their availability may be verified. A quantitative description of new or modified facilities and equipment shall be provided in sufficient detail to permit planning for construction or procurement.

Tailoring Guidelines:

See tailoring guidelines for paragraph 3.3.

3.3.3 Supply

This subparagraph shall be numbered 3.3.3 and shall describe the supply system, the impact of system requirements on the supply system, and the influence of the supply system on system design and use. This subparagraph shall include:

a. Introduction of new items into the supply system.

b. Re-supply methods.

c. Distribution and location of system stocks.

Tailoring Guidelines:

See tailoring guidelines for paragraph 3.3.

3.3.4 Government agencies

This paragraph shall be numbered 3.3.4 and shall identify the Government organizations that will be the development, support, and user agencies for the system.

Tailoring Guidelines:

See tailoring guidelines for paragraph 3.3.

3.4 System architecture

This paragraph shall be numbered 3.4 and shall describe the internal structure of the system. The segments, HWCIs, and CSCIs shall be identified and their purpose summarized. The relationships among the segments, HWCIs, and CSCIs shall be described. This paragraph shall also identify and state the purpose of each external interface of the system. A system architecture diagram may be used to illustrate the system top-level architecture.

Tailoring Guidelines:

As mentioned previously, the CSCIs identified in the SSDD DID should be defined as closely related sets of software components or individual high level software components. For a project using OOD, the CSCIs could be high level objects or related groups of objects. A preliminary version of this paragraph should be produced during the Preliminary System Analysis phase; this will provide guidance to the development of the initial system prototype, and subsequent prototypes. This paragraph should be expanded as system prototypes are evolved into a full-capability prototype. A final version will be written after the full-capability system prototype is completed and will be included in the definitized System Description, which is a major workproduct of the System Architecture phase. Where used, standard interfaces should be identified and described. It is also recommended that any diagrams in this paragraph use a nomenclature which identifies reusable components and standard interfaces.

3.5 Operational scenarios

This paragraph shall be numbered 3.5 and shall describe each operational scenario of the system. For each system state and mode, this paragraph shall identify the configuration items that execute and the manual operations to be performed. A table may be provided to illustrate the states and modes in which each configuration item executes and each manual operation is performed. In addition this paragraph shall describe the general flow of both execution control and data between configuration items while operating in the different states and modes. Flow diagrams may be used to illustrate execution control and data flow in each state and mode.

Tailoring Guidelines:

See tailoring guidelines for paragraph 3.4.

4. System design

This section shall be numbered 4 and shall be divided into the following paragraphs and subparagraphs to identify each HWCI, CSCI, and manual operation of the system. This section shall identify the HWCI(s) within the system that are to be designated as Prime item(s) or Critical items(s). A description of the relationship of HWCIs, CSCIs, and manual operations within the system shall be provided. A specification tree diagram(s) shall be used to describe the relationships between configuration items.

Tailoring Guidelines:

See tailoring guidelines for paragraph 3.4.

4.1 HWCI Identification

This paragraph shall be numbered 4.1 and shall be divided into subparagraphs to identify the system requirements allocated to each HWCI.

4.1.X (HWCI name and project-unique identifier)

This paragraph shall be numbered 4.1.X (beginning with 4.1.1), shall identify a HWCI by name and project-unique identifier, and shall state its purpose. This subparagraph shall identify each requirement from the System/Segment Specification allocated to the HWCI and the name and project-unique identifier of each system capability addressed by the HWCI. This subparagraph shall identify each interface external to the system addressed by the HWCI. Each interface external to the system shall be described in detailed quantitative terms (e.g., input/output voltages, dimensions, tolerances, loads, speeds, etc.). This subparagraph shall describe any design constraints on the HWCI.

Tailoring Guidelines:

Since a definitized System/Segment Specification will not be written until after the full-capability system prototype is developed, allocation of System/Segment Specification requirements to CIs will only be included in the definitized version of the System Description. Inclusion of quantitative definitions of external interfaces to CIs, prior to the definitized version, will be project dependent. In many cases, they will not be known until a definitized System/Segment Specification is available; at that time, the target hardware configuration will have been defined and the the full-capability system prototype developed.

4.2 CSCI Identification

This paragraph shall be numbered 4.2 and shall be divided into subparagraphs to identify the system requirements allocated to each CSCI.

4.2.X (CSCI name and project-unique identifier)

This subparagraph shall be numbered 4.2.X (beginning with 4.2.1), shall identify a CSCI by name and project-unique identifier, and shall state its purpose. This subparagraph shall identify each requirement from the System/Segment Specification allocated to the CSCI and the name and project-unique identifier of each system capability addressed by the CSCI. This subparagraph shall identify each interface external to the system addressed by the CSCI. Each interface external to the system shall be described in detailed quantitative terms (e.g., bits per second, word length, message

format, frequency of messages, priority rules, protocol). This subparagraph shall describe any design constraints on the CSCI.

Tailoring Guidelines:

See tailoring guidelines for subparagraph 4.2.X.

4.3 Manual operations identification

This paragraph shall be numbered 4.3 and shall be divided into subparagraphs to identify system requirements allocated to each manual operation.

4.3.X (Manual operation name and project-unique identifier)

This subparagraph shall be numbered 4.3.X (beginning with 4.3.1), shall identify a manual operation by name and project-unique identifier, and shall state its purpose. This subparagraph shall describe any design constraints that affect the manual operation and shall identify by name and project-unique identifier the capabilities from the System/Segment Specification to be satisfied by the manual operation.

Tailoring Guidelines:

Same as subparagraph 4.2.X, with regard to allocation of requirements.

4.4 Internal Interfaces

This paragraph shall be numbered 4.4 and shall be divided into the following subparagraphs to describe the interfaces that are internal to the system. This paragraph shall depict the relationship of the interfaces to the configuration items in the system. This subparagraph may reference a system internal interface diagram.

Tailoring Guidelines:

See tailoring guidelines for paragraph 3.4.

4.4.1 (HWCI-to-HWCI interface name and project-unique identifier)

This subparagraph shall be numbered 4.4.1 and shall identify by name and project-unique identifier all HWCI-to-HWCI interfaces within the system. This subparagraph shall identify each signal transmitted between HWCIs, the HWCI transmitting the signal, and the HWCI receiving the signal.

Tailoring Guidelines:

Only preliminary definitions, if at all, of these internal CI-to-CI interfaces, may be available for inclusion in this subparagraph prior to the development of the full-capability system prototype (this, of course, will be project dependent). In many cases, final definitions will not be known until after the full-capability system prototype is developed and a definitized System/Segment Specification is available. At that time, they will be included in the definitized System Description.

4.4.2 (HWCI-to-CSCI interface name and project-unique identifier)

This subparagraph shall be numbered 4.4.2 and shall specify by name and project-unique identifier all HWCI-to-CSCI interfaces within the system. This subparagraph shall identify each signal transmitted between a CSCI and an HWCI, the HWCI transmitting the signal, and the HWCI or CSCI receiving the signal.

Tailoring Guidelines:

See tailoring guidelines for paragraph 4.4.1.

4.4.3 (CSCI-to-CSCI interface name and project-unique identifier)

This subparagraph shall be numbered 4.4.3 and shall specify by name and project-unique identifier all CSCI-to-CSCI interfaces within the system. This subparagraph shall identify each data item transmitted between CSCIs, the CSCI transmitting the data, and the CSCI receiving the data.

Tailoring Guidelines:

As with the above internal HWCI-to-HWCI interface subparagraph, only preliminary definitions of the CSCI-to-CSCI interfaces will normally be provided in this subparagraph of the System Description, prior to the development of the full-capability system prototype (this, of course, will be project dependent). However, because definitions of internal CSCI-to-CSCI (component-to-component) interfaces are needed to build the initial and expanded system prototypes, the preliminary definitions should have enough detail to allow these prototypes to be built. Final definitions will be written after the full-capability system prototype is completed and will be included in the definitized version of the System Description.

5. Processing resources

This section shall be numbered 5 and shall be divided into the following paragraphs to describe the processing resources for the system.

5.X (Processing resource name and project-unique identifier)

This paragraph shall be numbered 5.X (beginning with 5.1) and shall identify a processing resource by name and project-unique identifier. This paragraph shall identify the configuration items, that use the resource. For each processing resource, this paragraph shall specify the hardware, programming, design, coding, and utilization characteristics of the processing resource. In addition, this paragraph shall define the following computer hardware characteristics of the processing resource, as applicable.

a. **Memory size.** Amount of internal memory (absolute, spare, or both) of the computer.

b. **Word size.** Number of bits in each computer word.

c. **Processing speed.** Computer processor capability (absolute, spare, or both) (e.g., a twenty percent reserve when in the full operational configuration).

d. **Character set standard.** Character set standard (e.g., ASCII, EBCDIC).

e. **Instruction set architecture.** Instruction set architecture.

f. **Interrupt capabilities.** Interrupt capabilities of the hardware.

g. **Direct Memory Access (DMA).** Data transfer by DMA.

h. **Channel requirements.** Channels and channel capacities (absolute, spare, or both).

i. **Auxiliary storage.** Auxiliary storage capacities (absolute, spare, or both).

j. **Growth capabilities.** Growth capability of any part of the processing resource.

k. **Diagnostic capabilities.** Diagnostic capabilities.

l. **Additional computer hardware capabilities.** Any additional computer hardware capabilities not previously mentioned (e.g., fault tolerance, preprocessing, floating point, array processor).

m. **Processing resource allocation.** The allocation of pertinent processing resources to each CSCI.

Tailoring Guidelines:

Only preliminary definitions of the processing resources will normally be provided in this paragraph of the System Description, prior to the development of the full-capability system prototype (this, of course, will be project dependent). However, these definitions of processing resources should have enough detail to allow the initial and expanded system prototypes to be built. Final definitions will be written after the full-capability system prototype is completed and will be included in the definitized version of the System Description.

6. Quality factor compliance

This section shall be numbered 6 and shall be divided into paragraphs and subparagraphs, as appropriate, to specify the models (and associated evaluation criteria) to be used to measure compliance with quality factor requirements.

Tailoring Guidelines:

Quality factor compliance will be included in the definitized version of the System Description, which is written after the full-capability system prototype has been completed. As appropiate, preliminary descriptions may be included in this paragraph.

7. Requirements traceability

This section shall be numbered 7 and shall provide traceability of the requirements allocated to the HWCIs, CSCIs, and manual operations back to the requirements of the System/Segment Specification. The traceability may be shown in a requirements traceability matrix.

Tailoring Guidelines:

Since a definitized System Specification will not be written until after the full-capability system prototype is developed, formal traceability of requirements will only be included in the definitized version of the System Description. However, some traceability should be included in this section between the PCS capabilities and the preliminary/expanded versions of the System Description.

Appendix C. Prototype Capability Specification

This appendix describes how the DoD-STD-2167A CDRL item DID (Data Item Description), for the System/Segment Specification (SSS), can be tailored for use as the SFLC Prototype Capability Specification (SS).

Differences Between DoD-STD-2167A and SFLC:

The System/Segment Specification (SSS), as defined in SSS DID, "specifies the requirements for a system or a segment of a system. Upon Government approval, and authentication, the SSS becomes the Functional Baseline for the system or segment." The System/Segment Specification (SSS), on a typical DoD-STD-2167A project, is produced and authenticated prior to the start of the system/software development effort, when the requirements and capabilities needed in the system (or segment) may not be well-understood and subject to change.

In the SFLC, the Prototype Capability Specification (PCS) is an evolving document which is used to describe the capabilities to be developed in initial or expanded versions of system or component prototypes. The PCS is also used to request the development of a new technology. The preliminary version of the PCS, which is produced during the Preliminary System Analysis phase, describes the initial system prototype. The PCS is expanded during the System Architecture phase, as the system prototypes evolve into a full-capability prototype. The SSS DID is well-suited for tailoring to the PCS as it contains sections which can provide a high-level description of a prototype, and its capabilities and interfaces.

General Tailoring Guidelines:

The capabilities referred to in the SSS DID will be used to define the prototype capabilities. As appropiate, capabilities should be related the reusable components. In addition, prototype level diagrams should provide a nomenclature for distinguishing between those capabilities which will come from the reusable component repository and those which will be newly developed. Where used, standard interfaces should also be identified and described. Since the PCS is a document which will evolve as prototype capabilities are expanded, sections of the PCS should clearly distinguish between previous prototype capabilities/interfaces and new capabilities/interfaces. Much of the information describing physical hardware-oriented characteristics (e.g., protective coatings) will not be required to be defined in the PCS.

The level of detail included in the definition of interfaces to external hardware or software systems will be project-dependent. In many cases, they will not be known until a definitized System Specification is available; at that time, the target hardware configuration will be defined and the full-capability system prototype developed. However, sufficient quantitative detail should be present to provide direction to the design and implementation of the prototype. After the full-capability system prototype has been completed, the information contained in the PCS will be used, as appropiate, in producing the definitized System Specification.

For a PCS which describes a component prototype, similar guidelines, as described above, would apply. SRS/IRS DIDs could be used instead of an SSS DID if more detailed information needed to be included in a PCS for a component prototype. The SRS/IRS tailoring guidelines described for a Component Specification could then be used in creating the component-level PCS.

For those projects using an object-oriented design (OOD) methodology, capabilities could be high level objects or related groups of objects.

Specific Tailoring Guidelines:

The DID text for each section is provided below, followed by specific tailoring guidelines for that section, as appropiate. The tailoring described is focused on what is required to allow the SSS DID to be used for the SFLC PCS. Additional tailoring of each section will be needed, to make it reflect the unique requirements of the system and the specific software development methodologies being used.

DoD-STD-2167A DID: System/Segment Specification

IDENTIFICATION NUMBER: DI-CMAN-80008A

Note: The word "system" used in the DID text that follows can mean either a system or a segment, as applicable.

3. System requirements

This section shall be numbered 3 and shall be divided into the following paragraphs and subparagraphs to specify the requirements for the system to which this specification applies.

3.1 Definition

This paragraph shall be numbered 3.1 and shall provide a brief description of the system. This description shall address pertinent operational, and logistical considerations and concepts. A system diagram shall be provided.

Tailoring Guidelines:

As mentioned previously, since the PCS is a document which will evolve as the prototype capabilities are expanded, this paragraph, and the paragraphs that follow, should also distinguish between those capabilities addressed by the previous prototype (if there is one) and those capabilities being addressed by the new prototype to be developed. The rationale for including the new capabilities in the prototype should also be included. Thus, the prototype description in this paragraph may draw much of its content from the capabilities already present and evaluated in the previous prototype. The system diagram will show the relationships between the prototype and external systems.

Note: In the paragraphs and subparagraphs that follow, the same kinds of distinctions and associated rationale between previous and new prototypes should also be made as appropiate. However, to allow for simplicity in the tailoring guidelines, these distinctions will be assumed and, thus, will not always be mentioned.

3.2 Characteristics

This paragraph shall be numbered 3.2 and shall be divided into the following subparagraphs to describe the requirements for system performance and physical characteristics.

3.2.1 Performance characteristics

This subparagraph shall be numbered 3.2.1 and shall be divided into the following subparagraphs to specify the system's capabilities in the context of the states in which the system can exist and the modes of operation within each state. Each capability of the system shall be specified in a uniquely identified subparagraph in order to provide for objective qualification.

3.2.1.X (State name): This subparagraph shall be numbered 3.2.1.X (beginning with 3.2.1.1) and shall identify and provide a brief description of a state in which the system can exist (e.g., weapon idle, weapon ready, weapon deployed).

Tailoring Guidelines:

In describing the prototype states, those states that were not part of the previous prototype should be identified.

3.2.1.X.Y (Mode name): This subparagraph shall be numbered 3.2.1.X.Y beginning with 3.2.1.1.1). This subparagraph shall identify and provide a brief description of a mode of operation (e.g., surveillance, threat evaluation, weapon assignment, target designation and acquisition, fire control resolution) within the system state identified above.

Tailoring Guidelines:

In describing the prototype modes, those modes that were not part of the previous prototype should be identified.

3.2.1.X.Y.Z (System capability name and project unique identifier): This subparagraph shall be numbered 3.2.1.X.Y.Z (beginning with 3.2.1.1.1.1), shall specify a capability of the system by name and project unique identifier, and shall describe its purpose. This subparagraph shall also identify the applicable parameters associated with the capability and shall express them in measurable terms. If a capability of a mode has been previously defined, this subparagraph shall reference rather than duplicate that information.

Tailoring Guidelines:

A prototype capability may be a reusable component used without modification, a reusable component which has been modified, or a capability to be developed. For those components which are reusable, reference to associated component libraries and documentation should be provided. For a project using object-oriented methods, the capabilities could be high level objects or related groups of objects.

3.2.2 System capability relationships

This subparagraph shall be numbered 3.2.2 and shall summarize the relationships between system capabilities and the states and modes of system.

Tailoring Guidelines:

None

3.2.3 External Interface requirements

This paragraph shall be numbered 3.2.3 and shall be divided into the following subparagraphs to describe requirements for interfaces with other systems. Detailed quantitative interface requirements may be defined in separate specifications or Interface Control Documents (ICDs) and referenced herein. All referenced ICDs are considered part of this specification.

3.2.3.X (System name) external interface description: This subparagraph shall be numbered 3.2.3.X (beginning with 3.2.3.1) and shall identify an external system with which this system interfaces. This subparagraph shall describe the interfaces to the external system. This subparagraph shall identify the purpose of each interface and shall describe the relationship between each interface and the states and modes of the system. When possible, each interface shall be specified in detailed, quantitative terms (e.g., dimensions, tolerances, loads, speeds, communications protocol).

Tailoring Guidelines: Where used, standard interfaces should be identified and associated documents referenced.

3.2.4 Physical characteristics

This subparagraph shall be numbered 3.2.4 and shall specify the requirements for the physical characteristics (e.g., weight limits, dimensional limits) of the system. Additional considerations for determining physical requirements include:

a. Transportation and storage

b. Security

c. Durability

d. Safety

e. Vulnerability

f. Color

Tailoring Guidelines:

For initial and intermediate prototypes most, if not all, of the requirements associated with this paragraph will not be defined in the PCS. However, as the system prototype expands to a full-capability prototype, these requirements may need to be included.

3.2.4.1 Protective coatings: This subparagraph shall be numbered 3.2.4.1 and shall specify, if applicable, protective coating requirements to assure protection from corrosion, abrasion, or other deleterious action.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4.

3.2.5 System quality factors

This paragraph shall be numbered 3.2.5 and shall be divided into the following subparagraphs to specify the applicable requirements pertaining to system quality factors.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware.

3.2.5.1 Reliability: This subparagraph shall be numbered 3.2.5.1, shall specify reliability requirements in quantitative terms, and shall define the conditions under which the reliability requirements are to be met. This subparagraph may include a reliability apportionment model to support apportionment of reliability values assigned to system capabilities for their share in achieving desired system reliability.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware. Software reliability requirements should be included as appropiate.

3.2.5.2 Maintainability: This subparagraph shall be numbered 3.2.5.2 and shall specify quantitative maintainability requirements. The requirements shall apply to maintenance in the planned maintenance and support environment and shall be stated in quantitative terms. Examples are:

a. Mean and maximum down time, reaction time, turnaround time, mean and maximum times to repair, mean time between maintenance actions.

b. Maximum effort required to locate and fix an error.

c. Maintenance man-hours per flying hour, maintenance man-hours per specific maintenance action, operational ready rate, maintenance hours per operating hour, frequency of preventative maintenance.

d. Number of people and skill levels, variety of support equipment.

e. Maintenance costs per operating hour, man-hours per overhaul.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware. Software maintainability requirements should be included as appropiate.

3.2.5.3 Availability: This subparagraph shall be numbered 3.2.5.3 and shall specify the degree to which the system shall be in an operable and committable state at the start of the mission(s), where the mission(s) is called for at an unknown (random) point in time.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware. Software availability requirements should be included as appropiate.

3.2.5.4 Additional quality factors: This subparagraph shall be numbered 3.2.5.4. and shall specify system quality requirements not defined in the above subparagraphs (e.g., integrity, efficiency, or correctness requirements of the system).

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware. Additional system quality requirements should be included as appropiate.

3.2.6 Environmental conditions

This paragraph shall be numbered 3.2.6 and shall specify the environmental conditions that the system must withstand during transportation, storage, and operation, such as:

a. Natural environment (e.g., wind, rain, temperature, geographic location)

b. Induced environment (e.g., motion, shock, noise, electromagnetic radiation)

c. Environments due to enemy action (e.g., over-pressure, explosions, radiation).

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware.

3.2.7 Transportability

This subparagraph shall be numbered 3.2.7 an shall specify any special requirements for transportation and materials handling. In addition, all system elements that, due to operational or functional characteristics, will be unsuitable for normal transportation methods shall be identified.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware.

3.2.8 Flexibility and expansion

This subparagraph shall be numbered 3.2.8 and shall specify areas of growth which require planning for system flexibility and expansion. In addition, this subparagraph shall specify specific system elements which require spare capacity to support flexibility and expansion.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to h dware. Software flexibility and expansion requirements should be included as appropiate.

3.2.9 Portability

This subparagraph shall be numbered 3.2.9 and shall specify requirements for portability which are applicable to the system to permit employment, deployment, and logistic support.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware. Software portability requirements should be included as appropiate.

3.3 Design and construction

This paragraph shall be numbered 3.3 and shall be divided into subparagraphs that specify minimum system design and construction standards which have general applicability to system equipment and are applicable to major classes of equipment (e.g. aerospace vehicle equipment, and support equipment) or are applicable to particular design standards. To the maximum extent possible, these requirements shall be specified by incorporation of the established military standards and specifications. Requirements which add to, but do not conflict with, requirements specified herein may be included in individual configuration item specifications. In addition, this paragraph shall specify criteria for the selection and imposition of Federal, military, and contractor specifications and standards.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware.

3.3.1 Materials

This subparagraph shall be numbered 3.3.1 and shall specify those system-peculiar requirements governing use of materials, parts, and processes in the design of system equipment. Special attention shall be directed to prevent unnecessary use of strategic or critical materials. (A strategic and critical materials list may be obtained from the contracting agency.) In addition, requirements, for the use of standard and commercial parts and parts for which qualified products lists have been established shall be specified in this paragraph.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware. Those repositories of reusable software components, which will be used in building the prototype, should also be identified.

3.3.1.1 Toxic products and formulations: This subparagraph shall be numbered 3.3.1.1 and shall specify requirements for the control of toxic products or formulations to be used in the system or to be generated by the system.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware.

3.3.2 Electromagnetic radiation

This subparagraph shall be numbered 3.3.2 and shall contain requirements pertaining to limits on the electromagnetic radiation which the system is permitted to generate.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware.

3.3.3 Nameplates and product marking

This subparagraph shall be numbered 3.3.3 and shall contain requirements for nameplates, part marking, serial and lot number marking, software media marking, and other identifying markings required for the system. Reference may be made to existing standards on the content and application of markings.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware. If multiple prototypes are being developed concurrently, requirements for identifying them should be included in this subparagraph.

3.3.4 Workmanship

This subparagraph shall be numbered 3.3.4 and shall specify workmanship requirements for equipment to be produced during system development and requirements for manufacture by production techniques.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware. Techniques being employed to produce high-reliability software could be included in this suparagraph.

3.3.5 Interchangeability

This subparagraph shall be numbered 3.3.5 and shall specify the requirements for system equipment to be interchangeable and replaceable. Entries in this paragraph are for the purpose of establishing a condition for design and are not to define the conditions of interchangeability required by the assignment of a part number.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware.

3.3.6 Safety

This subparagraph shall be numbered 3.3.6 and shall specify those safety requirements which are basic to the design of the system, with respect to equipment characteristics, methods of operation, and environmental influences. This paragraph shall also specify those safety requirements which prevent personnel injury and equipment degradation without degrading operational capability (e.g., restricting the use of dangerous materials where possible, classifying explosives for purposes of shipping, handling and storing, abort/escape provisions from enclosures, gas detection and warning devices, grounding of electrical system, cleanliness and decontamination, explosion proofing).

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware.

3.3.7 Human engineering

This subparagraph shall be numbered 3.3.7 and shall specify human engineering requirements for the system or for specific configuration items. This paragraph shall reference applicable documents (e.g., MIL-STD-1472) and specify any special or unique requirements (e.g., constraints on allocation of capabilities to personnel and communications, and personnel/equipment interactions). This paragraph shall include those specific areas, stations, or equipment which would require concentrated human engineering attention due to the sensitivity of the operation or criticality of the task; i.e., those areas where the effects of human error would be particularity serious.

Tailoring Guidelines:

For those prototypes requiring a human-computer interface, usability guidelines should be provided in this subparagraph for the user interface. The user needs (or parts of them), as defined in the System Description, which are being addressed by the prototype being developed should be clearly summarized in this subparagraph.

3.3.8 Nuclear control

This subparagraph shall be numbered 3.3.8 and shall specify system requirements for nuclear components, such as:

a. Component design

b. In-flight control

c. Prevention of inadvertent detonation

d. Nuclear safety rules.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware.

3.3.9 System security

This subparagraph shall be numbered 3.3.9 and shall specify security requirements that are basic to the design of the system with respect to the operational environment of the system. This subparagraph shall also specify those security requirements necessary to prevent compromise of sensitive information or materials.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware.

3.3.10 Government furnished property usage

This subparagraph shall be numbered 3.3.10 and shall specify any Government Furnished Equipment (GFE) to be incorporated into the system design. In addition, this paragraph shall specify any Government Furnished Information (GFI) and Government Furnished Software (GFS) to be incorporated into the system. This list shall identify the Government furnished property by reference to its nomenclature, specification number, and/or part number. If the list is extensive, it may be included as an appendix to this specification and referenced in this paragraph.

Tailoring Guidelines:

Government furnished reusable components, and the repositories from which they came, should be identified in this subparagraph.

3.3.11 Computer resource reserve capacity

This subparagraph shall be numbered 3.3.11 and shall specify the required computer resource reserve capacity (e.g., memory, timing, etc.).

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware. Reserve capacities required for prototype evolution and expansion could be included in this subparagraph.

3.4 Documentation

This paragraph shall be numbered 3.4 and shall specify the requirements for system documentation such as specifications, drawings, technical manuals, test plans and procedures, and installation instruction data.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware. Special documentation requirements that are needed for the prototype should be included in this paragraph.

3.5 Logistics

This paragraph shall be numbered 3.5 and shall specify logistic considerations and conditions that apply to the operational requirements. These considerations and conditions may include:

a. Maintenance

b. Transportation modes

c. Supply-system requirements

d. Impact on existing facilities

e. Impact on existing equipment.

Tailoring Guidelines:

Same as guidelines for paragraph 3.2.4 that relate to hardware.

3.6 Personnel and training

This paragraph shall be numbered 3.6 and be divided into the following subparagraphs to specify the requirements for personnel and training.

3.6.1 Personnel

This subparagraph shall be numbered 3.6.1 and shall specify personnel requirements which must be integrated into system design. These requirements shall be stated in terms of numbers plus tolerance and shall be the basis for contractor design and development decisions. Requirements stated in this paragraph shall be the basis for determination of system personnel training, training equipment, and training facility requirements. Personnel requirements shall include:

a. Numbers and skills of support personnel for each operational deployment mode and the intended duty cycle, both normal and emergency.

b. Skills and numbers of personnel that shall be allocated to the operation, maintenance, and control of the system.

Tailoring Guidelines:

For those prototypes requiring a human-computer interface, the numbers and skills of personnel required to use or support the prototype should be included in this subparagraph.

3.6.2 Training

This subparagraph shall be numbered 3.6.2 and shall include the following training requirements:

a. Contractor and Government responsibility for training. This For those prototypes requiring a human-computer interface, the numbers and skills of personnel required to use or support the prototype should be included in this subparagraph. subparagraph shall also specify the concept of how training shall be accomplished (e.g., school, contractor training).

b. Equipment that will be required for training purposes.

c. Training devices to be developed, characteristics of the training devices, and training and skills to be developed through the use of training devices.

d. Training time and locations available for a training program.

e. Source material and training aids to support the specified training.

Tailoring Guidelines:

For those prototypes requiring a human-computer interface, the training requirements for using or supporting the prototype should be included in this subparagraph.

3.7 Characteristics of subordinate elements

This paragraph shall be numbered 3.7 and shall be divided into the following subparagraphs to identify and describe each segment of the system. This subparagraph shall describe the relationships between the segments.

3.7.X (Segment name and project unique identifier)

This subparagraph shall be numbered 3.7.X (beginning with 3.7.1) and shall provide the following information for the segment:

a. State the purpose of the segment

b. Provide a brief description of the segment

c. Identify the system capabilities the segment performs.

Tailoring Guidelines:

If required, the subparagraphs below can be used to partition the system prototype into segments which will normally consist of well-defined groups of prototype capabilities. A segment may be a reusable segment used without modification, a reusable segment which needs to be modified, a segment which contains reusable components, or a new segment which has to be developed. For those segments or components which are reusable, reference to associated documentation should be provided. For a project using object-oriented methods, a segment could be a high level object or related groups of objects.

3.8 Precedence

This paragraph shall be numbered 3.8 and shall either specify the order of precedence of the requirements or assign weights to indicate the relative importance of the requirements.

Tailoring Guidelines:

None

3.9 Qualification

This paragraph shall be numbered 3.9 and shall state the requirements for verification or validation, as applicable, of capabilities in a specific application. Each qualification test shall be identified in a separate subparagraph and the specific application shall be described. Requirements shall be included for the conditions of testing, the time (program phase) of testing, period of testing, number of items to be tested, and any other pertinent qualification requirements.

Tailoring Guidelines:

This paragraph should describe the requirements for demonstration and evaluation of the prototype.

3.10 Standard example

This paragraph shall be numbered 3.10 and, if applicable, shall describe requirements for the production of one or more standard samples. Standard samples shall be limited to the illustration of qualities and characteristics that cannot be described using detailed test procedures or design data or that cannot be definitively expressed.

Tailoring Guidelines:

None

3.11 Preproduction sample, periodic production sample, or pilot lot

This paragraph shall be numbered 3.11 and, if applicable, shall describe requirements for producing a preproduction or periodic production sample, a pilot model, or a pilot lot.

Tailoring Guidelines:

None

4. Quality assurance provisions

This section shall be numbered 4 and shall be divided into the following paragraphs to specify the requirements to show how the requirements of section 3 and 5 shall be satisfied.

4.1 Responsibility for inspection

This paragraph shall be numbered 4.1 and shall assign responsibilities for performance of inspections of delivered products, materials, or services for determining compliance with all specified requirements.

Tailoring Guidelines:

None

4.2 Special tests and examinations

This paragraph shall be numbered 4.2 and shall specify any special tests and examinations required for sampling, lot formation, qualification evaluation, and any other tests or examinations as necessary. Each test and examination shall be described in a separate subparagraph.

Tailoring Guidelines:

None

4.3 Requirements cross reference

This paragraph shall be numbered 4.3 and shall correlate each system requirement in sections 3 and 5 to the quality assurance provisions specified in section 4. This paragraph may reference a requirements cross reference table which may be provided as an appendix to this specification.

Tailoring Guidelines:

None

5. Preparation for delivery

This section shall be numbered 5 and shall specify requirements for the preparation of the system and all its components for delivery, including packaging and handling. This section shall include requirements to document any non-standard practices in appropriate system and item specifications. This section may impose requirements to comply with standard practice by referencing ap-

propriate military specifications and standards to be used as the basis for preparing Section 5 of each specification for system end items.

Tailoring Guidelines:

If applicable, this section should include any special delivery requirements associated with the demonstration and operation of the prototype.

6. Notes

This section shall be numbered 6 and shall contain any general information that aids in understanding this document (e.g., background information, glossary). This section shall contain an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document.

Tailoring Guidelines:

None

6.1 Intended use

This paragraph shall be numbered 6.1 and shall briefly state the purpose of the system to which the SSS applies in terms of the mission and threat addressed by the system.

6.1.1 Missions

This subparagraph shall be numbered 6.1.1 and shall describe the missions of the system to the extent that such missions affect design requirements. This description shall include operational information, such as tactics, system deployment, operating locations, and facilities.

Tailoring Guidelines:

The primary and secondary mission(s) (or parts of them), as defined in the System Description, which are being addressed by the prototype being developed should be clearly summarized in this subparagraph, along with a rationale for the approach taken in addressing the missions.

6.1.2 Threat

This subparagraph shall be numbered 6.1.2 and shall describe the characteristics of potential targets, the characteristics of current and potential enemy weapon capabilities relevant to the system, and any additional threat considerations that affect the system design. This information may be contained in a separate document and referenced in this subparagraph if it is classified.

Tailoring Guidelines:

The threat (or parts of it), as defined in the System Description, which is being addressed by the prototype being developed should be clearly summarized in this subparagraph, along with a rationale for the approach taken in addressing the threat.

Appendix D. Prototype Design

This appendix describes how the DoD-STD-2167A CDRL item DID (Data Item Description), for the Software Design Document (SDD) and the Interface Design Document (IDD), can be tailored for use as the SFLC Prototype Design (PD).

Differences Between DoD-STD-2167A and SFLC:

The Software Design Document (SDD), as defined in SDD DID, "describes the complete design of a Computer Software Configuration Item (CSCI). It describes the CSCI as composed of Computer Software Components (CSCs) and Computer Software Units (CSUs." The Interface Design Document (IDD), as defined in IDD DID, "specifies the detailed design for one or more interfaces between one or more Computer Software Configuration Items (CSCIs) and other configuration items or critical items." The SDDs and IDDs, on a typical DoD-STD-2167A project, are produced and authenticated prior to the start of any system software build effort.

In the SFLC, the Prototype Design (PD) document is an evolving document which is used to describe the design of initial or expanded versions of system or component prototypes. A preliminary version of the system-level PD is produced during the System Architecture phase, to describe the design of the initial system prototype, and is expanded as the system prototypes evolve into a full-capability prototype. A preliminary version of a component-level PD is produced during the Software Growing phase for each system component to be developed. It describes the design of an initial component prototype, and is expanded as the prototype evolves into a full-capability component prototype.

The level of design detail in the PD will be less than that contained in a typical SDD and IDD, although it must be sufficient to provide enough information for the developers to build the prototype. For many system development efforts, an IDD will not need to be included as part of the PD. The actual level of detail required will be project and application dependent. The tailoring guidelines, described below, are focused on what is required to allow the SDD and IDD DIDs to be used for the SFLC PD.

General Tailoring Guidelines:

As mentioned previously, an SDD and IDD describe the design and interfaces of a CSCI. The SDD also describes the design of the associated CSCs (including sub-level CSCs) and CSUs. For a PD associated with a system prototype, the information, at the CSCI-level in the SDD and IDD DIDs, will be used to describe its design and interfaces. The design information, at the CSC and CSU-level in the SDD DID, will be used to describe the design of closely related sets of software subcomponents and/or individual high level software subcomponents (the choice will, of course, depend on the specific application). In this way, the design information in the SDD DID can be used to describe the design of each software component, which is part of a system prototype.

Note: A system developer can choose to allow CSCs and CSUs in an SDD to represent either a logical or physical software organization (JLC89). CSCs are intended to be "collections of CSUs and represent a higher-level abstraction of the requirements implemented by the CSUs" (JLC89).

For a PD which describes the design and interfaces of a component prototype, similar guidelines, as described above, would apply except that the term CSCI in the SDD and IDD DIDs would refer to the component prototype. CSCs and CSUs in the SDD DID would then be associated with lower-level subcomponents of the component prototype, and their associated interfaces.

A means should be provided within PDs for distinguishing between those components (or subcomponents) which will come from the reusable component repository and those which will be newly developed. For those components which come from the reusable component repository, reference should be made to the associated Component Design document for detailed design information. Diagrams, which are identified in the SDD and IDD DIDs, should also include a nomenclature for distinguishing between reusable components and components to be developed. Since the PD is a document which will evolve as prototype capabilities are expanded, sections of the PD should also clearly distinguish between the previous prototype design and the new prototype design.

For those projects using an object-oriented design (OOD) methodology, components or subcomponents should be high level objects or related groups of objects. Component or subcomponent interfaces, identified in the PD, would then define the flow of messages (operations) between them. Descriptions of CSU local data elements, as defined in the SDD DID, would be used to define encapsulated data within objects. Diagrams, associated with the specific object-oriented design methodology being used, would replace, as appropiate, the diagrams identified in the SDD and IDD DIDs.

Specific Tailoring Guidelines:

The DID text for each section is provided below, followed by specific tailoring guidelines for that section, as appropiate. The tailoring described is focused on what is required to allow the SDD and IDD DIDs to be used for the SFLC PD. Additional tailoring of each section will be needed, to make it reflect the unique requirements of the system and the specific software development methodologies being used.

DoD-STD-2167A DID: Software Design Document

IDENTIFICATION NUMBER: DI-MCCR-80012A

3. Preliminary design

This section shall be numbered 3 and shall be divided into the following paragraphs to describe the preliminary design of the CSCI.

3.1 CSCI overview

This paragraph shall be numbered 3.1 and shall identify and describe the role of the CSCI within the system to which this SDD applies. The overview shall identify and state the purpose of each external interface of the CSCI. A system architecture diagram may be used to show the relationships between this CSCI and the other CIs in the system.

Tailoring Guidelines:

For a PD which describes a system prototype, this paragraph will describe the role of the prototype relative to the development of the full-capability prototype. For a PD which describes a component prototype, this paragraph will describe the role of the component within the system. This paragraph will also identify and state the purpose of external interfaces of the prototype. Where used, standard interfaces should be identified and described. A prototype architecture diagram may be used to illustrate the prototype top-level architecture. Those components or subcomponents which will come from the reusable component repository, and those which will be newly developed, should be identified.

3.1.1 CSCI architecture

This paragraph shall be numbered 3.1.1 and shall describe the internal organizational structure of the CSCI. The Computer Software Components (CSCs) and sub-level CSCs shall be identified

and their purpose summarized. The relationships among the CSCs shall be described. The relationship description shall identify and state the purpose of each CSC-to-CSC interface and shall summarize the data transmitted via the interface. This paragraph shall identify any non-developmental software to be incorporated into the CSCI. The CSCI top-level architecture may be illustrated graphically.

Tailoring Guidelines:

The prototype architecture will be described in this paragraph. For a system prototype, high-level CSCs will be prototype components and sub-level CSCs will be associated subcomponents. For a component prototype, high-level CSCs will be prototype subcomponents and sub-level CSCs will be lower-level subcomponents. Thus, this paragraph will contain the purpose of the prototype components and subcomponents and the relationships among the components and subcomponents. This paragraph will also include descriptions of interfaces between components and subcomponents. Where used, standard interfaces should be identified and described. A prototype architecture diagram may be used to illustrate the prototype top-level architecture. Non-developmental software (i.e., COTS software), those components which will come from the reusable component repository, and those which will be newly developed should also be identified. For a project using OOD, CSCs will normally be collections of objects or individual objects.

Note: The system architecture will be finalized after the full-capability system prototype is completed, and will be included as part of the definitized System Description, which is a major workproduct of the System Architecture phase.

3.1.2 System states and modes

This paragraph shall be numbered 3.1.2 and shall identify each system state and mode in which the CSCI operates and the CSCs that execute in each state and mode. A state/CSC table may be provided to illustrate the system states and modes that each CSC executes. In addition, this paragraph shall describe the general flow of both execution control and data between CSCs while operating in the different states and modes. A flow diagram(s) may be used to illustrate the execution control and data flow in each state and mode.

Tailoring Guidelines:

The states and modes, addressed by the prototype, should be described in this section at the component or subcomponent levels.

3.1.3 Memory and processing time allocation

This paragraph shall be numbered 3.1.3 and shall document the allocation of memory and processing time to the CSCs. The allocation may be illustrated by a memory/processing time table (see Table 1).

CSC NAME	CSC NUM-BER	MEM .RY BUDGET (WORDS)	ALLOCATED PROC-ESSING TIME
MODE CONTROL	25	1,700	128.0 ms
COORDINATE CON-VERSION	69	900	155.0 ms 156.0 ms
RADAR CONTROL	26	3,000	96.0 ms
WEAPON CONTROL	27	2,100	100.0 ms
TARGET ENGAGEABILITY	11	1,700	10.0 ms
EXECUTIVE	1	1,200	80.0 ms
DATA BASE	100	2,000	N/A
TOTAL		12,600	570 ms
TOTAL AVAILABLE		16,384	740 ms
RESERVE		3,784	170 ms
RESERVE(%)		23	23

Table 1. Example of a CSC memory/processing time table

Tailoring Guidelines:

If appropiate, a preliminary allocation of memory and processing time to prototype components and subcomponents can be provided in this section. These will be refined as the prototype evolves and matures.

3.2 CSCI design description

This section shall be numbered 3.2 and shall be divided into the following subparagraphs to provide a design description of each CSC of the CSCI.

3.2.X (CSC name and project unique identifier)

This subparagraph shall be numbered 3.2.X (beginning with 3.2.1), shall identify a CSC by name and project unique identifier, and shall state its purpose. This subparagraph shall provide the following information:

a. Identify the requirements allocated to the CSC from the applicable requirements specification(s). If the CSC is composed of sub-level CSCs, some or all of this information may be referenced and provided by the sub-level CSC description.

b. Describe the preliminary design of the CSC in terms of execution control and data flow. If a CSC is composed of sub-level CSCs, this description shall identify the relationships among the sub-level CSCs. In addition this description shall identify each CSCI internal interface documented in the Software Requirements Specification, that is to be addressed by the CSC and its sub-level CSCs, as applicable. This information may be referenced rather than duplicated for each sub-level CSC.

c. Identify the derived design requirements for the CSC and any design constraints imposed on or by the CSC. If the CSC is composed of sub-level CSCs, some or all of this information may be referenced and provided by the sub-level CSC description.

Tailoring Guidelines:

Those requirements allocated to the CSC (component or subcomponent) from the associated Prototype Capabililities Specification(s) should be identified, along with a rationale for the allocation. As appropiate, execution control, data flow, internal interfaces, derived design requirements,

and design constraints will also be described in this paragraph at the component or subcomponent levels.

3.2.X.Y (Sub-level CSC name and project unique identifier): This subparagraph shall be numbered 3.2.X.Y (beginning with 3.2.1.1), shall identify a sub-level CSC by name and project unique identifier, shall state it s purpose, and shall provide the information required by a through c above. This subparagraph does not apply if there are no sub-level CSCs. If this CSC is also composed of sub-level CSCs, each sub-level CSC shall be identified by name and project unique identifier and the information required by a through c above shall be provided in a separate subparagraph for each sub-level CSC.

Tailoring Guidelines:

The same tailoring guidelines described above for Section 3.2.X are also applicable to this subparagraph.

4. Detailed design

This section shall be numbered 4 and shall be divided into the following paragraphs and subparagraphs to describe the detailed design of each CSC.

4.X (CSC name and project unique identifier)

This paragraph shall be numbered 4.X (beginning with 4.1), and shall be divided into the following subparagraphs to identify and describe each of the Computer Software Units (CSUs) of a CSC. This paragraph shall describe the relationships of the CSUs in terms of execution control and data flow between the CSUs of this CSC and shall identify all CSU interfaces that are external to the CSC. Each CSU that is used by more than one CSC shall be described in detail under one CSC and then referenced by the other using CSCs.

Tailoring Guidelines:

The CSUs described in this paragraph can represent the logical or physical organization of prototype components or subcomponents (CSCs). If the CSUs represent the physical organization, then they will normally be Ada library units, i.e., package specifications and bodies, procedures, functions, or tasks. Those CSUs which will come from the reusable component repository, and those which will be newly developed, should be identified. For reusable CSUs, reference should be made to the associated software design documentation. If appropiate, execution control, data flow, and external interfaces will be described at the CSU level. Where used, standard interfaces should be identified and described, as required.

4.X.Y (CSU name and project unique identifier)

This subparagraph shall be numbered 4.X.Y (beginning with 4.1.1) and shall identify a CSU by name and project unique identifier and shall state the purpose of the CSU. This subparagraph shall be divided into the following subparagraphs to provide the design information for the CSU.

Tailoring Guidelines:

Those CSUs which will come from the reusable component repository, and those which will be newly developed, should be identified. For reusable CSUs, reference should be made to the associated software design documentation.

4.X.Y.1 (CSU name) Design specification/constraints: This subparagraph shall be numbered 4.X.Y.1 (beginning with 4.1.1.1) and shall state the design requirements for the CSU. This subparagraph shall identify the requirements allocated to the CSC that are to be satisfied or partially satisfied by the CSU and shall identify any constraints on the design of the CSU. The design requirements addressed in this subparagraph shall include design requirements for the man-machine interface, as applicable.

Tailoring Guidelines:

Those CSC requirements satisfied or partially satisfied by the CSU (subcomponent) should be identified, along with a rationale for the allocation. As appropiate, design requirements and design constraints will also be described in this subparagraph.

4.X.Y.2 (CSU name) Design: This subparagraph shall be numbered 4.X.Y.2 (beginning with 4.1.1.2) and shall specify the design of the CSU. If the CSU is to be coded in a programming language other than the specified CSCI language, the programming language shall be identified and the rationale for its use shall be provided. If the CSU resides in a library, this subparagraph shall identify the library by name and project unique identifier, and the design document in which the library description can be found. The detailed design information identified below shall be provided for the CSU, as applicable. This information may be provided by automated tools or other techniques, such as a program design language, flowcharts, or other design representations.

a. Input/output data elements. Identify and state the purpose of each input and output data element to the CSU. The design information for data elements shall be provided in section 5.

b. Local data elements. Identify and state the purpose of each data element that originates in the CSU and is not used by any other CSU. Each data element shall be described in terms of name, brief description, data type, data representation, size, units of measure, limit/range, accuracy, precision/resolution, and any other attributes of the data. This information may be provided in a CSU local data definition table.

c. Interrupts and signals. Identify and describe the interrupts and signals handles by the CSU. Identify for each interrupt and signal, as appropriate, its source, purpose, priority, expected response and response time, and minimum, maximum, and probable frequency of occurrence.

d. Algorithms. Identify, state the purpose, and describe in detail the algorithms to be incorporated into the execution of the CSU. The algorithms shall be described in terms of the manipulation of input and local data elements and the generation of output data elements.

e. Error handling. Identify and describe the error detection and recovery features of the CSU, including handling of erroneous input data and other conditions that effect the execution of the CSU.

f. Data conversion. Identify and describe any data conversion operations performed in order to implement the CSU's interfaces.

g. Use of other elements. Describe the use of other elements that are used by the CSU including, but not limited to:

    1) Other CSUs (e.g., calls for library functions, calls for I/O services for access to databases, mass storage devices, and real-time I/O channels).

    2) Shared data stored in a global memory (e.g., databases or data files, tables, compool, datapool, etc.).

    3) Input and output buffers, including message buffers.

h. Logic flow. Describe the logic flow of the CSU in terms of items "a" through "g" above. Describe the conditions under which CSU execution is initiated and, if applicable, communication interface features are invoked, and the conditions under which control is passed to other CSUs, as applicable, If sequencing is dynamically controlled during the CSCI's operations, the method for sequence control and the logic and input conditions of that method shall be described, such as timing variations, priority assignments, internal operations such as data transfer in and out of internal memory, sensing of discrete input signals, and timing relationships between interrupt operations with the CSCI.

i. Data structures. Describe local data structures implemented by the CSU and any shared data structures used by the CSU. Shared data structures shall be described under one CSU and referenced thereafter by the sharing CSUs.

j. Local data files or database. If a data file(s) or a database are part of the local data of a CSU, state the purpose of each file or database, the structure of each file or database in terms of records, fields, etc., and describe the access procedures, such as sequential or random.

k. Limitations. Describe any limitations or unusual features that restrict the performance of the CSU.

Tailoring Guidelines:

For those CSUs which will come from the reusable component repository, the repository should be identified and reference made to the associated software design documentation. For those which will be newly developed, design information in this paragraph should be at a sufficient level of detail to allow the CSU to be implemented.

5. CSCI data

This section shall be numbered 5 and shall describe the global data elements within the CSCI. For ease in readability and maintenance, the information required below may be provided in one or more tables. The following information shall be provided for each data element, as applicable:

a. For data elements internal to the CSCI:

1) Name of the data element

2) A brief description

3) The units of measure, such as knots, seconds, meters, feet, etc.

4) The limit/range of values required for the data element (for constants provide the actual value)

5) The accuracy required for the data element

6) The precision/resolution in terms of significant digits

7) For real time systems, the frequency at which the data element is calculated or refreshed such as 10 KHz, 50 Msec, etc.

8) Legality checks performed on the data element

9) The data type, such as integer, ASCII, fixed, real, enumeration, etc.

10) The data representation/format

11) The CSU project unique identifier where the data element is set or calculated

12) The CSU project unique identifier(s) where the data element is used

13) The data source from which the data is supplied, such as database or data file, global common, local common, compool, datapool, parameter, etc. Where applicable, each source shall be identified by its project unique identifier.

b. For data elements of the CSCI's external interfaces:

1) Identify the data element

2) Identify the interface by name and project unique identifier

3) Reference the Interface Design Document (IDD) in which the external interface is described.

Tailoring Guidelines:

This subparagraph will describe data elements internal and external to the system or component prototype. The data element descriptions should contain sufficient detail to support the implementation of the system or component prototype.

6. CSCI data files

This section shall be numbered 6 and shall be divided into the following paragraphs to describe each of the shared data files of the CSCI.

6.1 Data file to CSC/CSU cross reference

This paragraph shall be numbered 6.1 and shall provide a mapping of each data file identified below to the CSCs and CSUs that use the data file.

Tailoring Guidelines:

This paragraph will provide a mapping of the data files to the prototype components and subcomponents that use the data file. For a project using OOD, the data files represent the encapsulated data of the prototype objects.

6.X (Data file name and project unique identifier)

This subparagraph shall be numbered 6.X (beginning with 6.2) and shall identify by name and project unique identifier a data file of the CSCI that is shared by more than one CSU. This paragraph shall state the purpose of the data file, identify the maximum size of the file, and describe the file access method, such as random or sequential. This paragraph shall provide a description of the structure and size of the records contained within the file. This paragraph shall also provide a description of the data that is to reside in the file. The data description shall include, as applicable, data type, data representation, size, units of measure, limit/range, accuracy, precision/resolution, and any other design characteristics of the data. This information may be provided in a file definition table.

Tailoring Guidelines:

For a project using OOD, the data files represent the encapsulated data of the prototype objects. For encapsulated data within an object that is shared by more than one CSU, this paragraph should describe the data at a sufficient level of detail to support the implementation of the system or component prototype.

7. Requirements traceability

This section shall be numbered 7 and shall provide traceability of the requirements allocated down to the CSU level of each CSC back to the requirements of the Software Requirements Specification and Interface Requirements Specification. The traceability may be shown graphically.

Tailoring Guidelines:

Traceability of requirements, allocated down to the CSU level of each CSC, back to the requirements of the associated Prototype Capabilility Specification should be provided in this section.

DoD-STD-2167A DID: Interface Design Document

IDENTIFICATION NUMBER: DI-MCCR-80027A

3. Interface design

This section shall be numbered 3 and shall be divided into the following paragraphs and subparagraphs to describe the interface design for those interfaces to which this IDD applies.

Tailoring Guidelines:

Only preliminary definitions of prototype interfaces will normally be provided in a PD, prior to the development of a full-capability prototype (this, of course, will be project dependent). Final definitions will be written after the full-capability prototype is completed and will be included in definitized versions of the system and component design documents, as required.

3.1 Interface diagrams

This paragraph shall be numbered 3.1 and shall specify for each CSCI to which this IDD applies, its relationship to the other HWCIs, CSCIs, or critical items with which it interfaces. This description may be provided by one or more interface diagrams, as appropriate.

Tailoring Guidelines:

For a system prototype, this paragraph will show the relationships among the prototype components and subcomponents, as well as to other hardware or software components to which it interfaces. For a component prototype, this paragraph will show the relationships to other hardware or software components to which it interfaces. Where used, standard interfaces should be identified.

3.X (Interface name and project-unique identifier)

This paragraph shall be numbered 3.X (beginning with 3.2), shall identify an interface by name and project-unique identifier, and shall state its purpose. This paragraph shall be divided into the following subparagraphs to describe the design of the interface.

Tailoring Guidelines:

This paragraph, and the subparagrahs that follow, shall provide the preliminary design of the prototype interfaces These preliminary definitions should have enough detail to allow the associated prototype to be built (the level of detail will be project dependent).

3.X.1 Data elements

This subparagraph shall be numbered 3.X.1 (beginning with 3.2.1) and shall provide, in a data element definition table, the following information, as applicable, for each data element transmitted across the interface:

a. A project-unique identifier for the data element

b. A brief description of the data element

c. The CSCI, HWCI, or critical item that is the source of the data element

d. The CSCI(s), HWCI(s), or critical item(s) that are the users of the data element

e. The units of measure required for the data element, such as seconds, meters, kilohertz, etc.

f. The limit/range of values required for the data element (for constants provide the actual value)

g. The accuracy required for the data element

h. The precision or resolution required for the data element in terms of significant digits.

i. The frequency at which the data element is calculated or refreshed, such as 10 KHz or 50 Msec

j. Legality checks performed on the data element

k. The data type, such as integer, ASCII, fixed, real, enumerated, etc.

l. The data representation/format

m. The priority of the data element.

3.X.2 Message descriptions

This subparagraph shall be numbered 3.X.2 (beginning with 3.2.2), shall identify each message transmitted across the interface by name and project-unique identifier, and shall describe the assignment of data elements to each message. A cross-reference of each message to the data elements that embody the message shall be provided. In addition, a cross-reference of each data element to the message(s) of which it is a part shall also be provided. Cross-references may be provided as an appendix and referenced in this subparagraph.

3.X.3 Interface priority

This subparagraph shall be numbered 3.X.3 (beginning with 3.2.3) and shall specify the relative priority of the interface and of each message transmitted across the interface.

3.X.4 Communications protocol

This subparagraph shall be numbered 3.X.4 (Beginning with 3.2.4) and shall be divided into the following subparagraphs to describe the commercial, military, or proprietary communications protocols associated with the interface.

3.X.4.Y (Protocol name): This subparagraph shall be numbered 3.X.4.Y (beginning with 3.2.4.1), shall identify a protocol by name and shall describe the technical details of the protocol. This subparagraph shall address the following communications specification details, as applicable:

a. Fragmentation and reassembly of messages

b. Message formatting

c. Error control and recovery procedures, including fault tolerance features

d. Synchronization, including connection establishment, maintenance, termination, and timing

e. Flow control, concluding sequence numbering, window size, and buffer allocation

f. Data transfer rate, whether it is periodic or aperiodic, and minimum interval between transfers

g. Routing, addressing, and naming conventions

h. Transmission services, including priority and grade

i. Status, identification, notification, and any other reporting features

j. Security, including encryption, user authentication, compartmentalization and auditing.

Appendix E. System Specification

This appendix describes how the DoD-STD-2167A CDRL item DID (Data Item Description), for the System/Segment Specification (SSS), can be tailored for use as the SFLC System Specification (SS).

Differences Between DoD-STD-2167A and SFLC:

The System/Segment Specification (SSS), as defined in SSS DID, "specifies the requirements for a system or a segment of a system. Upon Government approval, and authentication, the SSS becomes the Functional Baseline for the system or segment." The System/Segment Specification (SSS), on a typical DoD-STD-2167A project, is produced and authenticated prior to the start of the system/software development effort, when the requirements and capabilities needed in the system (or segment) may not be well-understood and subject to change.

In the SFLC, the System Specification (SS) in contrast to the SSS, is produced after a full-capability prototype has been developed. At this point the requirements and capabilities of the system are well-known, and can be defined in very definitive terms. In addition, the partitioning of the system into components has been accomplished, reusable components integrated, and new components developed.

General Tailoring Guidelines:

It is recommended that the system (or segment) capabilities, referred to in the SSS DID, be as closely related to the already defined system components, as is feasible. This will allow specification of interface and performance requirements, and other associated information, in the SSS DID to also be related to components.

Specific Tailoring Guidelines:

The DID text for each section is provided below, followed by specific tailoring guidelines for that section, as appropiate. The tailoring described is focused on what is required to allow the SSS DID to be used for the SFLC SS. Additional tailoring of each section will be needed, to make it reflect the unique requirements of the system and the specific software development methodologies being used.

DoD-STD-2167A DID: System/Segment Specification

IDENTIFICATION NUMBER: DI-CMAN-80008A

Note: The word "system" used in the DID text that follows can mean either a system or a segment, as applicable.

3. System requirements

This section shall be numbered 3 and shall be divided into the following paragraphs and subparagraphs to specify the requirements for the system to which this specification applies.

3.1 Definition

This paragraph shall be numbered 3.1 and shall provide a brief description of the system. This description shall address pertinent operational, and logistical considerations and concepts. A system diagram shall be provided.

Tailoring Guidelines:

The system description in this paragraph will draw much of its content from the capabilities already present and evaluated in the full-capability prototype. The system diagram will show the relationships between the system and external systems.

3.2 Characteristics

This paragraph shall be numbered 3.2 and shall be divided into the following subparagraphs to describe the requirements for system performance and physical characteristics.

3.2.1 Performance characteristics

This subparagraph shall be numbered 3.2.1 and shall be divided into the following subparagraphs to specify the system's capabilities in the context of the states in which the system can exist and the modes of operation within each state. Each capability of the system shall be specified in a uniquely identified subparagraph in order to provide for objective qualification.

3.2.1.X (State name): This subparagraph shall be numbered 3.2.1.X (beginning with 3.2.1.1) and shall identify and provide a brief description of a state in which the system can exist (e.g., weapon idle, weapon ready, weapon deployed).

Tailoring Guidelines:

None

3.2.1.X.Y (Mode name): This subparagraph shall be numbered 3.2.1.X.Y beginning with 3.2.1.1.1). This subparagraph shall identify and provide a brief description of a mode of operation (e.g., surveillance, threat evaluation, weapon assignment, target designation and acquisition, fire control resolution) within the system state identified above.

Tailoring Guidelines:

None

3.2.1.X.Y.Z (System capability name and project unique identifier): This subparagraph shall be numbered 3.2.1.X.Y.Z (beginning with 3.2.1.1.1.1), shall specify a capability of the system by name and project unique identifier, and shall describe its purpose. This subparagraph shall also identify the applicable parameters associated with the capability and shall express them in measurable terms. If a capability of a mode has been previously defined, this subparagraph shall reference rather than duplicate that information.

Tailoring Guidelines:

Most of the capabilities of the system described in this paragraph have already been implemented and evaluated in the full-capability prototype. It is recommended that the system (or segment) capabilities be as closely related to the already defined system component, as is feasible. A component may be a reusable component, used without modification; a reusable component, which has been modified; or a newly developed component. For those components which are reusable, reference to associated documentation should be provided. For a project using object-oriented methods, the capabilities could be high level objects or related groups of objects.

3.2.2 System capability relationships

This subparagraph shall be numbered 3.2.2 and shall summarize the relationships between system capabilities and the states and modes of system.

Tailoring Guidelines:

None

3.2.3 External Interface requirements

This paragraph shall be numbered 3.2.3 and shall be divided into the following subparagraphs to describe requirements for interfaces with other systems. Detailed quantitative interface requirements may be defined in separate specifications or interface Control Documents (ICDs) and referenced herein. All referenced ICDs are considered part of this specification.

3.2.3.X (System name) external interface description: This subparagraph shall be numbered 3.2.3.X (beginning with 3.2.3.1) and shall identify an external system with which this system interfaces. This subparagraph shall describe the interfaces to the external system. This subparagraph shall identify the purpose of each interface and shall describe the relationship between each interface and the states and modes of the system. When possible, each interface shall be specified in detailed, quantitative terms (e.g., dimensions, tolerances, loads, speeds, communications protocol).

Tailoring Guidelines:

Where used, standard interfaces should be identified and associated documents referenced.

3.2.4 Physical characteristics

This subparagraph shall be numbered 3.2.4 and shall specify the requirements for the physical characteristics (e.g., weight limits, dimensional limits) of the system. Additional considerations for determining physical requirements include:

a. Transportation and storage

b. Security

c. Durability

d. Safety

e. Vulnerability

f. Color

Tailoring Guidelines:

None

3.2.4.1 Protective coatings: This subparagraph shall be numbered 3.2.4.1 and shall specify, if applicable, protective coating requirements to assure protection from corrosion, abrasion, or other deleterious action.

Tailoring Guidelines:

None

3.2.5 System quality factors

This paragraph shall be numbered 3.2.5 and shall be divided into the following subparagraphs to specify the applicable requirements pertaining to system quality factors.

3.2.5.1 Reliability: This subparagraph shall be numbered 3.2.5.1, shall specify reliability requirements in quantitative terms, and shall define the conditions under which the reliability requirements are to be met. This subparagraph may include a reliability apportionment model to support apportionment of reliability values assigned to system capabilities for their share in achieving desired system reliability.

Tailoring Guidelines:

None

3.2.5.2 Maintainability: This subparagraph shall be numbered 3.2.5.2 and shall specify quantitative maintainability requirements. The requirements shall apply to maintenance in the planned maintenance and support environment and shall be stated in quantitative terms. Examples are:

a. Mean and maximum down time, reaction time, turnaround time, mean and maximum times to repair, mean time between maintenance actions.

b. Maximum effort required to locate and fix an error.

c. Maintenance man-hours per flying hour, maintenance man-hours per specific maintenance action, operational ready rate, maintenance hours per operating hour, frequency of preventative maintenance.

d. Number of people and skill levels, variety of support equipment.

e. Maintenance costs per operating hour, man-hours per overhaul.

Tailoring Guidelines:

None

3.2.5.3 Availability: This subparagraph shall be numbered 3.2.5.3 and shall specify the degree to which the system shall be in an operable and committable state at the start of the mission(s), where the mission(s) is called for at an unknown (random) point in time.

Tailoring Guidelines:

None

3.2.5.4 Additional quality factors: This subparagraph shall be numbered 3.2.5.4. and shall specify system quality requirements not defined in the above subparagraphs (e.g., integrity, efficiency, or correctness requirements of the system).

Tailoring Guidelines:

None

3.2.6 Environmental conditions

This paragraph shall be numbered 3.2.6 and shall specify the environmental conditions that the system must withstand during transportation, storage, and operation, such as:

a. Natural environment (e.g., wind, rain, temperature, geographic location)

b. Induced environment (e.g., motion, shock, noise, electromagnetic radiation)

c. Environments due to enemy action (e.g., over-pressure, explosions, radiation).

Tailoring Guidelines:

None

3.2.7 Transportability

This subparagraph shall be numbered 3.2.7 an shall specify any special requirements for transportation and materials handling. In addition, all system elements that, due to operational or functional characteristics, will be unsuitable for normal transportation methods shall be identified.

Tailoring Guidelines:

None

3.2.8 Flexibility and expansion

This subparagraph shall be numbered 3.2.8 and shall specify areas of growth which require planning for system flexibility and expansion. In addition, this subparagraph shall specify specific system elements which require spare capacity to support flexibility and expansion.

Tailoring Guidelines:

None

3.2.9 Portability

This subparagraph shall be numbered 3.2.9 and shall specify requirements for portability which are applicable to the system to permit employment, deployment, and logistic support.

Tailoring Guidelines:

None

3.3 Design and construction

This paragraph shall be numbered 3.3 and shall be divided into subparagraphs that specify minimum system design and construction standards which have general applicability to system equipment and are applicable to major classes of equipment (e.g. aerospace vehicle equipment, and support equipment) or are applicable to particular design standards. To the maximum extent possible, these requirements shall be specified by incorporation of the established military standards and specifications. Requirements which add to, but do not conflict with, requirements specified herein may be included in individual configuration item specifications. In addition, this paragraph shall specify criteria for the selection and imposition of Federal, military, and contractor specifications and standards.

Tailoring Guidelines:

None

3.3.1 Materials

This subparagraph shall be numbered 3.3.1 and shall specify those system-peculiar requirements governing use of materials, parts, and processes in the design of system equipment. Special attention shall be directed to prevent unnecessary use of strategic or critical materials. (A strategic and critical materials list may be obtained from the contracting agency.) In addition, requirements, for the use of standard and commercial parts and parts for which qualified products lists have been established shall be specified in this paragraph.

Tailoring Guidelines:

None

3.3.1.1 Toxic products and formulations: This subparagraph shall be numbered 3.3.1.1 and shall specify requirements for the control of toxic products or formulations to be used in the system or to be generated by the system.

Tailoring Guidelines:

None

3.3.2 Electromagnetic radiation

This subparagraph shall be numbered 3.3.2 and shall contain requirements pertaining to limits on the electromagnetic radiation which the system is permitted to generate.

Tailoring Guidelines:

None

3.3.3 Nameplates and product marking

This subparagraph shall be numbered 3.3.3 and shall contain requirements for nameplates, part marking, serial and lot number marking, software media marking, and other identifying markings required for the system. Reference may be made to existing standards on the content and application of markings.

Tailoring Guidelines:

None

3.3.4 Workmanship

This subparagraph shall be numbered 3.3.4 and shall specify workmanship requirements for equipment to be produced during system development and requirements for manufacture by production techniques.

Tailoring Guidelines:

None

3.3.5 Interchangeability

This subparagraph shall be numbered 3.3.5 and shall specify the requirements for system equipment to be interchangeable and replaceable. Entries in this paragraph are for the purpose of establishing a condition for design and are not to define the conditions of interchangeability required by the assignment of a part number.

Tailoring Guidelines:

None

3.3.6 Safety

This subparagraph shall be numbered 3.3.6 and shall specify those safety requirements which are basic to the design of the system, with respect to equipment characteristics, methods of operation, and environmental influences. This paragraph shall also specify those safety requirements which prevent personnel injury and equipment degradation without degrading operational capability (e.g., restricting the use of dangerous materials where possible, classifying explosives for purposes of shipping, handling and storing, abort/escape provisions from enclosures, gas detection and warning devices, grounding of electrical system, cleanliness and decontamination, explosion proofing).

Tailoring Guidelines:

None

3.3.7 Human engineering

This subparagraph shall be numbered 3.3.7 and shall specify human engineering requirements for the system or for specific configuration items. This paragraph shall reference applicable documents (e.g., MIL-STD-1472) and specify any special or unique requirements (e.g., constraints on allocation of capabilities to personnel and communications, and personnel/equipment interactions). This paragraph shall include those specific areas, stations, or equipment which would require concentrated human engineering attention due to the sensitivity of the operation or criticality of the task; i.e., those areas where the effects of human error would be particularity serious.

Tailoring Guidelines:

None

3.3.8 Nuclear control

This subparagraph shall be numbered 3.3.8 and shall specify system requirements for nuclear components, such as:

a. Component design

b. In-flight control

c. Prevention of inadvertent detonation

d. Nuclear safety rules.

Tailoring Guidelines:

None

3.3.9 System security

This subparagraph shall be numbered 3.3.9 and shall specify security requirements that are basic to the design of the system with respect to the operational environment of the system. This subparagraph shall also specify those security requirements necessary to prevent compromise of sensitive information or materials.

Tailoring Guidelines:

None

3.3.10 Government furnished property usage

This subparagraph shall be numbered 3.3.10 and shall specify any Government Furnished Equipment (GFE) to be incorporated into the system design. In addition, this paragraph shall specify any Government Furnished Information (GFI) and Government Furnished Software (GFS) to be incorporated into the system. This list shall identify the Government furnished property by reference to its nomenclature, specification number, and/or part number. If the list is extensive, it may be included as an appendix to this specification and referenced in this paragraph.

Tailoring Guidelines:

Government furnished reusable components, and the repositories from which they came, should be identified in this subparagraph.

3.3.11 Computer resource reserve capacity

This subparagraph shall be numbered 3.3.11 and shall specify the required computer resource reserve capacity (e.g., memory, timing, etc.).

Tailoring Guidelines:

None

3.4 Documentation

This paragraph shall be numbered 3.4 and shall specify the requirements for system documentation such as specifications, drawings, technical manuals, test plans and procedures, and installation instruction data.

Tailoring Guidelines:

None

3.5 Logistics

This paragraph shall be numbered 3.5 and shall specify logistic considerations and conditions that apply to the operational requirements. These considerations and conditions may include:

a. Maintenance

b. Transportation modes

c. Supply-system requirements

d. Impact on existing facilities

e. Impact on existing equipment.

Tailoring Guidelines:

None

3.6 Personnel and training

This paragraph shall be numbered 3.6 and be divided into the following subparagraphs to specify the requirements for personnel and training.

3.6.1 Personnel

This subparagraph shall be numbered 3.6.1 and shall specify personnel requirements which must be integrated into system design. These requirements shall be stated in terms of numbers plus tolerance and shall be the basis for contractor design and development decisions. Requirements stated in this paragraph shall be the basis for determination of system personnel training, training equipment, and training facility requirements. Personnel requirements shall include:

a. Numbers and skills of support personnel for each operational deployment mode and the intended duty cycle, both normal and emergency.

b. Skills and numbers of personnel that shall be allocated to the operation, maintenance, and control of the system.

Tailoring Guidelines:

None

3.6.2 Training

This subparagraph shall be numbered 3.6.2 and shall include the following training requirements:

a. Contractor and Government responsibility for training. This subparagraph shall also specify the concept of how training shall be accomplished (e.g., school, contractor training).

b. Equipment that will be required for training purposes.

c. Training devices to be developed, characteristics of the training devices, and training and skills to be developed through the use of training devices.

d. Training time and locations available for a training program.

e. Source material and training aids to support the specified training.

Tailoring Guidelines:

None

3.7 Characteristics of subordinate elements

This paragraph shall be numbered 3.7 and shall be divided into the following subparagraphs to identify and describe each segment of the system. This subparagraph shall describe the relationships between the segments.

3.7.X (Segment name and project unique identifier)

This subparagraph shall be numbered 3.7.X (beginning with 3.7.1) and shall provide the following information for the segment:

a. State the purpose of the segment

b. Provide a brief description of the segment

c. Identify the system capabilities the segment performs.

Tailoring Guidelines:

More then likely, prototype segments of the system have already been implemented and evaluated in the full-capability prototype. Thus, segments will normally consist of well-defined groups of system capabilities. A segment may be a reusable segment used without modification, a reusable segment which has been modified, or a newly developed segment. For those segments which are reusable, reference to associated documentation should be provided. For a project using object-oriented methods, a segment could be a high level object or related groups of objects.

3.8 Precedence

This paragraph shall be numbered 3.8 and shall either specify the order of precedence of the requirements or assign weights to indicate the relative importance of the requirements.

Tailoring Guidelines:

None

3.9 Qualification

This paragraph shall be numbered 3.9 and shall state the requirements for verification or validation, as applicable, of capabilities in a specific application. Each qualification test shall be identified in

a separate subparagraph and the specific application shall be described. Requirements shall be included for the conditions of testing, the time (program phase) of testing, period of testing, number of items to be tested, and any other pertinent qualification requirements.

Tailoring Guidelines:

None

3.10 Standard example

This paragraph shall be numbered 3.10 and, if applicable, shall describe requirements for the production of one or more standard samples. Standard samples shall be limited to the illustration of qualities and characteristics that cannot be described using detailed test procedures or design data or that cannot be definitively expressed.

Tailoring Guidelines:

None

3.11 Preproduction sample, periodic production sample, or pilot lot

This paragraph shall be numbered 3.11 and, if applicable, shall describe requirements for producing a preproduction or periodic production sample, a pilot model, or a pilot lot.

Tailoring Guidelines:

Reference should be made in this paragraph to the full-capability prototype, the results of the prototype demonstration and evaluation, and the associated prototype documentation.

4. Quality assurance provisions

This section shall be numbered 4 and shall be divided into the following paragraphs to specify the requirements to show how the requirements of section 3 and 5 shall be satisfied.

4.1 Responsibility for inspection

This paragraph shall be numbered 4.1 and shall assign responsibilities for performance of inspections of delivered products, materials, or services for determining compliance with all specified requirements.

Tailoring Guidelines:

None

4.2 Special tests and examinations

This paragraph shall be numbered 4.2 and shall specify any special tests and examinations required for sampling, lot formation, qualification evaluation, and any other tests or examinations as necessary. Each test and examination shall be described in a separate subparagraph.

Tailoring Guidelines:

None

4.3 Requirements cross reference

This paragraph shall be numbered 4.3 and shall correlate each system requirement in sections 3 and 5 to the quality assurance provisions specified in section 4. This paragraph may reference a requirements cross reference table which may be provided as an appendix to this specification.

Tailoring Guidelines:

None

5. Preparation for delivery

This section shall be numbered 5 and shall specify requirements for the preparation of the system and all its components for delivery, including packaging and handling. This section shall include requirements to document any non-standard practices in appropriate system and item specifications. This section may impose requirements to comply with standard practice by referencing appropriate military specifications and standards to be used as the basis for preparing Section 5 of each specification for system end items.

Tailoring Guidelines:

None

6. Notes

This section shall be numbered 6 and shall contain any general information that aids in understanding this document (e.g., background information, glossary). This section shall contain an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document.

Tailoring Guidelines:

None

6.1 Intended use

This paragraph shall be numbered 6.1 and shall briefly state the purpose of the system to which the SSS applies in terms of the mission and threat addressed by the system.

Tailoring Guidelines:

None

6.1.1 Missions

This subparagraph shall be numbered 6.1.1 and shall describe the missions of the system to the extent that such missions affect design requirements. This description shall include operational information, such as tactics, system deployment, operating locations, and facilities.

Tailoring Guidelines:

None

6.1.2 Threat

This subparagraph shall be numbered 6.1.2 and shall describe the characteristics of potential targets, the characteristics of current and potential enemy weapon capabilities relevant to the system, and

any additional threat considerations that affect the system design. This information may be contained in a separate document and referenced in this subparagraph if it is classified.

Tailoring Guidelines:

None

Appendix F. Software Design

This appendix describes how the DoD-STD-2167A CDRL item DID (Data Item Description), for the Software Product Specification (SPS), can be tailored for use as the SFLC Software Design (SWD).

Differences Between DoD-STD-2167A and SFLC:

The Software Product Specification (SPS), as defined in SDD DID, "consists of the Software Design Document (SDD) and source code listings for a Computer Software Configuration Item (CSCI)." Also, as stated in the SPS DID, "Upon Government appproval and authentification following the Physical Configuration Audit (PCA), the SPS establishes the Product Baseline for the CSCI."

In the SFLC, multiple Software Design documents are produced in the Productization and Production phase, where the full-capability system prototype is converted into a fully-tested productized system. At that point the design and development of the system is completed, and each SWD provides definitized documentation of the design and source code for a software component of the system.

General Tailoring Guidelines:

As mentioned previously, an SPS contains the SDD and source code for a CSCI. A CSCI, within the SFLC, is equivalent to either a set of closely related software components and/or individual high level software components (the choices will depend on the specific application). Thus, in an SWD the information, identified at the CSCI-level in the SDD DID, will be used describe a system component. The productized system will consist of both reusable and newly developed software components. For a reusable component, an existing SWD, Component Design document, or source code listing can be referenced. For a newly developed component, the associated Component Design can be referenced or included in the SWD; however, the associated source code should be included in the SWD.

Specific Tailoring Guidelines:

The DID text for each section is provided below, followed by specific tailoring guidelines for that section, as appropiate. The tailoring described is focused on what is required to allow the SPS DID to be used for the SFLC SWD. Additional tailoring of each section will be needed, to make it reflect the unique requirements of the system and the specific software development methodologies being used.

DoD-STD-2167A DID: Software Product Specification

IDENTIFICATION NUMBER: DI-MCCR-80029A

3. Requirements

This section shall be numbered 3 and shall be divided into the following paragraphs to contain, or reference the appendixes that contain, all design documentation and listings applicable to the CSCI.

3.1 Software design

This paragraph shall be numbered 3.1 and shall contain, or reference the appendix or other document that contains, the Software Design Document (SDD).

Tailoring Guidelines:

As mentioned previously, a CSCI, within the SFLC, is equivalent to either a set of closely related software components and/or a high level software component (the choices will depend on the specific application). A component may be a reusable component, used without modification; a reusable component, which has been modified; or a newly developed component. For a reusable component, an existing SWD or Component Design document can be referenced. For a reusable component, which has been modified, or a newly developed component, the associated Component Design can be referenced or included in the SWD. For a project using object-oriented methods, a component could be a high level object or related groups of objects.

3.2 CSCI source code listings

This paragraph shall be numbered 3.2 and shall contain, or reference the appendix that contains, the source code listings of the CSCI. This paragraph shall provide an index that cross-references each CSC and CSU to the location in the source code listings where they are found.

Tailoring Guidelines:

For a reusable component, a source code listing can be referenced in an existing SWD. For a reusable component, which has been modified, or a newly developed component, the associated source code listing should be included in this paragraph or referenced in the SWD appendix.

3.3 Compiler/assembler

This paragraph shall be numbered 3.3 and shall specify the compiler and, if applicable, the assembler used to translate the source code.

Tailoring Guidelines:

None

3.4 Measured resource utilization

This paragraph shall be numbered 3.4 and shall specify the measured resource utilization of the CSCI at the time of delivery.

Tailoring Guidelines:

None

4. Notes

This section shall be numbered 4 and shall contain any general information that aids in understanding this specification (e.g., background information, glossary, formula derivations). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document.

Tailoring Guidelines:

None

Appendixes

Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendixes may be bound as separate documents for ease in handling. Appendixes shall be lettered alphabetically (A, B, etc.), and the paragraphs within each appendix be numbered as multiples of 10 (e.g., Appendix A, paragraph 10, 10.1, 10.2, 20, 20.1 20.2, etc.). Pages within each appendix shall be numbered alpha-numerically as follows: Appendix A pages shall be numbered A-1, A-2, A-3, etc. Appendix B pages shall be numbered B-1, B-2, B-3, etc.

Tailoring Guidelines:

None

Appendix A Software design

This appendix shall contain the SDD if that document is not contained in paragraph 3.1 or in another referenced document. If the SDD is included herein, the paragraph numbers and page numbers need not be changed to comply with the requirement stated in 10.2.7 of this DID.

Tailoring Guidelines:

See tailoring guidelines for paragraph 3.1.

Appendix B Source code listings

This appendix shall contain the source code listings of the CSCI if they are not contained in paragraph 3.2.

Tailoring Guidelines:

See tailoring guidelines for paragraph 3.2.

Appendix C Additional appendixes

Any additional appendixes shall start with Appendix C.

Tailoring Guidelines:

None

Appendix G. Component Specification

This appendix describes how the DoD-STD-2167A CDRL item DID (Data Item Description), for the Software Requirement Specification (SRS) and the Interface Requirement Specification (IRS), can be tailored for use as the SFLC Component Specification (CS).

Differences Between DoD-STD-2167A and SFLC:

The Software Requirements Specification (SRS), as defined in SRS DID, "specifies the engineering and qualification requirements for a Computer Software Configuration Item (CSCI)." The Interface Requirements Specification (IRS), as defined in IRS DID, "specifies the requirements for one or more interfaces between one or more Computer Software Configuration Item (CSCI) and other configuration items or critical items". The SRS and IRS on a typical DoD-STD-2167A project, are produced and authenticated prior to the software design and development effort.

In the SFLC, Component Specifications (CS), in contrast to the SRS and IRS, are produced while a full-capability system prototype is being developed. Thus, many of the hardware or software components to which the component, described by a CS, interfaces may be prototypes. A primary source of material used in creating the CS will be the Prototype Capability Specification (PCS), an evolving informal document which is used to describe the capabilities to be incorporated into a component prototype, as it evolves from an initial to a full-capability prototype.

General Tailoring Guidelines:

An SRS and IRS describes CSCI Requirements. A CSCI, within the SFLC, should be equivalent to either a set of closely related software components or individual high level software components (the choices will depend on the specific application). This will allow the requirements information, identified in the SRS DID, to be used to describe the requirements of each software component, which is part of a system. In the CS, a means should also be provided for distinguishing between those components or subcomponents which will come from the reusable component repository and those which will be newly developed. For those which come from the reusable component repository, reference should be made to the associated Component Specification for detailed requirements information. Diagrams, which are identified in the SRS and IRS DID, should also include a nomenclature for distinguishing between reusable subcomponents and subcomponents to be developed.

For those projects using an object-oriented design (OOD) methodology, capabilities, as described in the SRS DID, should be high level objects or related groups of objects in the CS. Interfaces in the CS would then define the flow of messages (operations) between objects. Descriptions of internal data elements, as described in the SRS DID, would be used to define encapsulated data within objects. Diagrams, associated with the specific object-oriented design methodology being used, would replace, as appropiate, the diagrams identified in the SRS DID.

Specific Tailoring Guidelines:

The DID text for each section is provided below, followed by specific tailoring guidelines for that section, as appropiate. The tailoring described is focused on what is required to allow the SRS and IRS DIDs to be used for the SFLC CS. Additional tailoring of each section will be needed, to make it reflect the unique requirements of the system and the specific software development methodologies being used.

DoD-STD-2167A DID: Software Requirement Specification (SRS)

IDENTIFICATION NUMBER: DI-MCCR-80025A

3.0 Engineering requirements

This section shall be numbered 3 and shall be divided into the following paragraphs and subparagraphs to specify the engineering requirements necessary to ensure proper development of the CSCI. Requirements to be included herein shall be allocated or derived from requirements established by the applicable SSS, PIDS, or CIDS.

Tailoring Guidelines:

In the SFLC, as previously mentioned, a CSCI can be defined as closely related sets of software components or individual high level software components. For a project using object-oriented methods, the CSCIs could be high level objects or related groups of objects. A component may be a reusable component, used without modification; a reusable component, which has been modified; or a newly developed component. A definitized System Specification will not be written until after the full-capability system prototype is developed. Since CSs will normally be developed while the full-capability system prototype is still under development, allocated or derived requirements from the System Specification cannot be included in the CS. However, requirements allocated or derived from the system-level Prototype Capability Specification (PCS) or associated component-level PCS(s) should be included.

3.1 CSCI external interface requirements

This paragraph shall be numbered 3.1 and shall identify the external interfaces of the CSCI. An external interface diagram similar to Figure 1 (in the referenced DID) may be used to aid in this description. Each external interface shall be identified by name and project-unique identifier and a brief description of each interface shall be provided. Any identifying documentation, such as an interface Control Document or Interface Requirements Specification, shall be referenced for each interface.

Tailoring Guidelines:

This paragraph will describe the external interfaces of this component. An external interface diagram may be used to show the relationships of this component to other hardware or software components to which it interfaces. Some of these components may be reusable components or prototypes which are part of a system prototype. Where used, standard interfaces should be identified. As appropiate, the associated interface documentation should be referenced.

3.2 CSCI capability requirements

This paragraph shall be numbered 3.2 and shall identify, in the subparagraphs that follow, all of the capability requirements that the CSCI must satisfy. If the system of which the CSCI is a part can exist in various system states and modes as documented in the system specification, this paragraph shall identify each such state and mode and shall correlate each CSCI capability to those states and modes. A table may be used to depict this correlation.

Tailoring Guidelines:

The states and modes, addressed by the component, should be described in this paragraph at the subcomponent level.

3.2.X (Capability name and project-unique identifier)

This subparagraph shall be numbered 3.2.X (beginning with 3.2.1), shall identify the CSCI capability by name and project-unique identifier and shall state the purpose of the capability and its performance in measurable terms. This subparagraph shall identify and state the purpose of each input and output associated with the capability. This subparagraph shall identify the allocated or derived requirements that the capability satisfies or partially satisfies. If the capability can be more clearly specified by decomposing it into constituent capabilities, the requirements for each constituent capability shall be provided as one or more subparagraphs. Each constituent capability shall be assigned a project-unique identifier that is derived from the identifier of the parent capability.

Tailoring Guidelines:

Where possible, capabilities of a CSCI should be related to reusable components, and reference made to the related requirements specifications. For capabilities which cannot be related to reusable components, this subparagraph will identify the requirements that the capability satisfies or partially satisfies, as defined in the DID instructions above. Additional requirements, not specified in the associated PCSs (e.g., error-checking and performance enhancements) should also be specified. For a project using object-oriented methods, capabilities would normally be collections of objects or individual objects.

3.3 CSCI internal interfaces

This paragraph shall be numbered 3.3 and shall identify the interfaces between the capabilities identified above. Each internal interface shall be identified by name and project-unique identifier and a brief description of each interface shall be provided, including a summary of the information transmitted over the interface. Internal interface diagrams depicting data flow, control flow, and other relevant information may be used to aid in this description.

Tailoring Guidelines:

This paragraph will describe the internal interfaces of this component. An internal interface diagram may be used to show the relationships between subcomponents. Where used, standard interfaces should be identified and the associated interface documentation referenced. For a project using object-oriented methods, internal interfaces would normally define messages, i.e., the operations performed by the objects, of which this component is composed.

3.4 CSCI data element requirement

This paragraph shall be numbered 3.4 and shall specify the information identified below, as applicable.

a. For data elements internal to the CSCI:

   1) Assign a project-unique identifier to the data elements

   2) Provide a brief description of the data element

   3) Identify the Units of measure required for the data element, such as seconds, meters, kilohertz, etc.

   4) Identify the limit/range of values required for the data element (for constants provide the actual value)

   5) Identify the accuracy required for the data element.

   6) Identify the precision or resolution required for the data element in terms of significant digits

   7) For data elements of the CSCI's internal interfaces:

      - Identify the interface by name and project-unique identifier

- Identify the source capability of the data element by name and project-unique identifier
- Identify the destination capability of the data element by name and project-unique identifier.

b. For data elements of the CSCI's external interfaces:

1) Identify the data elements by project-unique identifier

2) Identify the interface by name and project-unique identifier

3) Identify the source or destination capability, as applicable, by name and project-unique identifier

4) Reference the Interface Requirements Specification in which the interface is specified.

Tailoring Guidelines: This subparagraph will describe data elements internal and external to this component. The data element descriptions should contain sufficient detail to support the implementation of the component. For a project using object-oriented methods, the internal data elements would describe the elements of the encapsulated data of an object, and external data elements would describe the elements of messages (operations performed by an object) and the responses to those messages.

3.5 Adaptation requirements

This paragraph shall be numbered 3.5 and shall be divided into the following subparagraphs to specify the requirements for adapting the CSCI to site-unique conditions and to changes in the system environment.

Tailoring Guidelines:

The subparagraphs below will describe the installation-dependent data and operational parameters used by this component. The descriptions should contain sufficient detail to support the subsequent design of the component. For a project using object-oriented methods, the subparagraphs below would describe the data used in input messages (operations) and the objects in which this data is used.

3.5.1 Installation-dependent data

This subparagraph shall be numbered 3.5.1 and shall describe the site-unique data required by each installation. Examples of such data are: site latitude and longitude, radar ranges and areas of coverage, and prescribed safety limits. In addition, this subparagraph shall identify the CSCI capabilities in which these data are used.

3.5.2 Operational parameters

This subparagraph shall be numbered 3.5.2 and shall describe parameters required by the CSCI that may vary within a specified range according to operational needs. Examples of such data are: allowable trajectory deviations, navigation set model numbers, airplane performance characteristics, interaction/isolation of sorties, missile performance characteristics. This subparagraph shall identify the CSCI capabilities in which these data are used.

3.6 Sizing and timing requirements

This paragraph shall be numbered 3.6 and shall specify the amount and, if applicable, location of internal and auxiliary memory and the amount of processing time allocated to the CSCI. This paragraph shall specify the resources required of both memory and the central processing unit (CPU) for the CSCI.

Tailoring Guidelines:

If appropiate, a preliminary allocation of memory amount and processing time to this component can be provided in this section. For a project using object-oriented methods, an allocation to objects within the component could also be included.

3.7 Safety requirements

This paragraph shall be numbered 3.7 and shall specify safety requirements that are applicable to the design of the CSCI, with respect to potential hazards to personnel, property, and the physical environment.

Tailoring Guidelines:

None

3.8 Security requirements

This paragraph shall be numbered 3.8 and shall specify security requirements that are applicable to the design of the CSCI, with respect to potential compromise of sensitive data.

Tailoring Guidelines:

None

3.9 Design constraints

This paragraph shall be numbered 3.9 and shall specify other requirements that constrain the CSCI design, such as the use of a particular processing configuration, etc.

Tailoring Guidelines:

None

3.10 Software quality factors

This paragraphs shall be numbered 3.10 and shall be divided into subparagraphs, as appropriate, to specify each software quality factor identified in the contract or derived from a higher level specification. For each quality factor required, the method of compliance shall be specified along with the requirements for that factor.

Tailoring Guidelines:

None

3.11 Human performance/human engineering requirements

This paragraph shall be numbered 3.11 and shall specify the applicable human factors engineering requirements for the CSCI. These requirements shall include, as applicable, considerations for:

a. Human information processing capabilities and limitations

b. Foreseeable human errors under both normal and extreme conditions

c. Implications for the total system environment (include training, support, and operational environment).

Tailoring Guidelines:

None

3.12 Requirements traceability

This paragraph shall be numbered 3.12 and shall contain a mapping of the engineering requirements in this specification to the requirements applicable to this CSCI in the SSS, PIDS, or CIDS. This paragraph shall also provide a mapping of the allocation of the CSCI requirements from the SSS, PIDS, or CIDS to the engineering requirements in this specification.

Tailoring Guidelines:

A definitized System Specification will not be written until after the full-capability system prototype is developed. Since CSs will normally be developed while the full-capability system prototype is still under development, formal allocation and traceability between the System Specification requirements and engineering requirements in this CS cannot be provided in this paragraph. However, traceability and allocation of requirements between the system-level Prototype Capability Specification (PCS) or associated component-level PCS(s) could be included.

4. Qualification requirements

This section shall be numbered 4 and shall be divided into the following paragraphs to specify the qualification methods and any special qualification requirements necessary to establish that the CSCI satisfies the requirements of sections 3 and 5.

Tailoring Guidelines:

None

4.1 Qualification methods

a. **Demonstration.** The operation of the CSCI (or some part of the CSCI) that relies on observable functional operation not requiring the use of elaborate instrumentation or special test equipment.

b. **Analysis.** The processing of accumulated data obtained from other qualification methods. Examples are interpretation or extrapolation of test data.

c. **Inspection.** The virtual examination of CSCI code, documentation, etc.

REQUIREMENT NAME	IDENTIFIER	SECTION 3 PARAGRAPH	QUALIFICATION METHOD(S)*	QUAL LEVEL **	SECTION 4 FORMAL TEST PARAGRAPH
TRANSFER DATA	CAP 103-A	3.2.10.3	A	3	4.2.10
RECORD DATA	CAP 205-C	3.2.13.5	A,I	1	4.2.13

Table 2. Example of a qualification cross-reference table

* QUALIFICATION METHOD

A - ANALYSIS

D - DEMONSTRATION

I - INSPECTION

** QUALIFICATION LEVEL

1 - CONFIGURATION ITEM

2 - SYSTEM INTEGRATION

3 - SYSTEM

4 - SYSTEM INSTALLATION

Tailoring Guidelines:

None

4.2 Special qualification requirements

This paragraph shall be numbered 4.2 and shall be divided into appropriate subparagraphs to specify special requirements associated with qualification of the CSCI. This paragraph shall identify and describe, if applicable, special tools, techniques (e.g., test formulas, algorithms), procedures, facilities, and acceptance limits. For each special test the following information shall be specified:

a. A project-unique identifier for the test

b. The paragraph number(s) of the capability requirements(s) to which the test applies

c. A description of the test, such as peak-load stress test for 24 hr. duration

d. The level of the test (CSU, CSC, CSCI, segment, or system level).

Tailoring Guidelines:

None

DoD-STD-2167A DID: Interface Requirement Specification

IDENTIFICATION NUMBER: DI-MCCR-80026A

3. Interface specification

This section shall be numbered 3 and shall be divided into the following paragraphs and subparagraphs to specify the requirements for those interfaces to which this IRS applies.

3.1 Interface diagrams

This paragraph shall be numbered 3.1 and shall identify the interfaces among the CSCIs, HWCIs, and critical items to which this specification applies. One or more interface diagrams, as appropriate, shall be provided to depict the interfaces. Each interface shall be identified by name and project-unique identifier.

Tailoring Guidelines:

The interfaces depicted on external interface diagram(s) will identify directional flows of data between this component and other hardware and software components (i.e., there will be separately identified input and output interfaces between this component and other components). Some of

these components may be reusable components or prototypes which are part of a system prototype. Where used, standard interfaces should be identified. As appropiate, the associated interface documentation should be referenced.

3.X (Interface name and project-unique identifier)

This paragraph shall be numbered 3.X (beginning with 3.2), shall identify an interface by name and project-unique identifier, and shall state its purpose. This paragraph shall be divided into the following subparagraphs to specify the requirements for the interface and for the data transmitted across the interface.

Tailoring Guidelines:

None

3.X.1 Interface requirements

This subparagraph shall be numbered 3.X.1 (beginning with 3.2.1) and shall specify the following, as applicable:

a. Whether the interfacing CSCIs are to execute concurrently or sequentially. If concurrently, the method of inter-CSCI synchronization to be used.

b. The communication protocol to be used for the interface.

c. The priority level of the interface.

Tailoring Guidelines:

This paragraph will describe the external interfaces of this component. Some of these components may be reusable components or prototypes which are part of a system prototype. Where used, standard interfaces should be identified. As appropiate, the associated interface documentation should be referenced.

3.X.2 Data requirements

This subparagraph shall be numbered 3.X.2 (beginning with 3.2.2) and shall specify, in a data element definition table similar to Table I, the following information, as applicable, for each data element transmitted across the interface:

a. A project-unique identifier for the data element

b. A brief description of the data element

c. The CSCI, HWCI, or critical item that is the source of the data element

d. The CSCI(s), HWCI(s), or critical item(s) that are the users of the data element

e. The Units of measure required for the data element, such as seconds, meters, kilohertz, etc.

f. The limit/range of values required for the data element (for constants provide the actual value)

g. The accuracy required for the data element

h. The precision or resolution required for the data element in terms of significant digits.

Tailoring Guidelines:

The data definitions, provided in this subparagraph, should contain sufficient detail to support the subsequent design of the component interfaces. For a project using object-oriented methods, the

data element definitions would describe the contents of input messages (operations) and the responses to those messages.

Appendix H. Component Design

This appendix describes how the DoD-STD-2167A CDRL item DID (Data Item Description), for the Software Design Document (SDD) and the Interface Design Document (IDD), can be tailored for use as the SFLC Component Design (CD).

Differences Between DoD-STD-2167A and SFLC:

The Software Design Document (SDD), as defined in SDD DID, "describes the complete design of a Computer Software Configuration Item (CSCI). It describes the CSCI as composed of Computer Software Components (CSCs) and Computer Software Units (CSUs." The Interface Design Document (IDD), as defined in IDD DID, "specifies the detailed design of one or more interfaces between one or more Computer Software Configuration Items (CSCIs) and other configuration items or critical items." The SDDs and IDDs, on a typical DoD-STD-2167A project, are produced and authenticated prior to the start of any system software build effort.

In the SFLC, the Component Design (CD) document, in contrast to the SDD and IDD, is produced after a full-capability component prototype has been developed. At this point the design of the component, and its associated interfaces, are well-known and can be defined in very definitive terms. In addition, the partitioning of the component into subcomponents has been accomplished, reusable subcomponents identified, and new subcomponents developed. The design, described in the CD, will satisfy the capabilities and requirements of the component, as described in the Component Specification (CS). A source of the material used in creating the CD will be the Prototype Design document, an evolving informal document which is used to describe the design of a component prototype, as it evolves from an initial to a full-capability prototype.

General Tailoring Guidelines:

An SDD and IDD describe the design and interfaces of a CSCI. The SDD also describes the design of the associated CSCs (including sub-level CSCs) and CSUs. In a CD, information identified at the CSCI-level in the SDD and IDD DIDs will be used to describe the design and interfaces of a component of a system. The design information, identified at the CSC and CSU-level in the SDD DID, will be used to describe the design of closely related sets of software subcomponents and/or individual software subcomponents (the choice will, of course, depend on the specific application).

A means should be provided in the CD for distinguishing between those subcomponents which will come from the reusable component repository and those which will be newly developed. For those subcomponents which come from the reusable component repository, reference should be made to the associated Component Design document for detailed design information. Diagrams, which are identified in the SDD and IDD DIDs, should also include a nomenclature for distinguishing between reusable subcomponents and subcomponents to be developed.

For those projects using an object-oriented design (OOD) methodology, subcomponents should be high level objects or related groups of objects. Subcomponent interfaces, identified in the CD, would then define the flow of messages (operations) between subcomponents. Descriptions of local data elements of CSUs, as defined in the SDD DID, would be used to define encapsulated data within objects. Diagrams, associated with the specific object-oriented design methodology being used, would replace, as appropiate, the diagrams identified in the SDD DID.

Specific Tailoring Guidelines:

The DID text for each section is provided below, followed by specific tailoring guidelines for that section, as appropiate. The tailoring described is focused on what is required to allow the SDD and

IDD DIDs to be used for the SFLC CD. Additional tailoring of each section will be needed, to make it reflect the unique requirements of the system and the specific software development methodologies being used.

DoD-STD-2167A DID: Software Design Document

IDENTIFICATION NUMBER: DI-MCCR-80012A

3. Preliminary design

This section shall be numbered 3 and shall be divided into the following paragraphs to describe the preliminary design of the CSCI.

3.1 CSCI overview

This paragraph shall be numbered 3.1 and shall identify and describe the role of the CSCI within the system to which this SDD applies. The overview shall identify and state the purpose of each external interface of the CSCI. A system architecture diagram may be used to show the relationships between this CSCI and the other CIs in the system.

Tailoring Guidelines:

In the SFLC, a CSCI can be defined as closely related sets of software components or individual high level software components. For a project using OOD, the CSCIs could be high level objects or related groups of objects. A component may be a reusable component, used without modification; a reusable component, which has been modified; or a newly developed component. Those components which will come from the reusable component repository, and those which will be newly developed, should be identified. This paragraph will describe the role of the component within the system. This paragraph will also identify and state the purpose of external interfaces of the component. Where used, standard interfaces should be identified and described. A system architecture diagram may be used to show the relationships between this component and other components in the system.

3.1.1 CSCI architecture

This paragraph shall be numbered 3.1.1 and shall describe the internal organizational structure of the CSCI. The Computer Software Components (CSCs) and sub-level CSCs shall be identified and their purpose summarized. The relationships among the CSCs shall be described. The relationship description shall identify and state the purpose of each CSC-to-CSC interface and shall summarize the data transmitted via the interface. This paragraph shall identify any non-developmental software to be incorporated into the CSCI. The CSCI top-level architecture may be illustrated graphically.

Tailoring Guidelines:.

The component architecture will be described in this paragraph. High-level CSCs will be high-level subcomponents and sub-level CSCs will be lower-level subcomponents. This paragraph will contain the purpose of the subcomponents and the relationships among the subcomponents. This paragraph will also include descriptions of interfaces between subcomponents. Where used, standard interfaces should be identified and described. A component architecture diagram may be used to illustrate the top-level architecture. Non-developmental software (i.e., COTS software), those components which will come from the reusable component repository, and those which will be newly developed should also be identified. For a project using OOD, CSCs will normally be collections of objects or individual objects.

3.1.2 System states and modes

This paragraph shall be numbered 3.1.2 and shall identify each system state and mode in which the CSCI operates and the CSCs that execute in each state and mode. A state/CSC table may be provided to illustrate the system states and modes that each CSC executes. In addition, this paragraph shall describe the general flow of both execution control and data between CSCs while operating in the different states and modes. A flow diagram(s) may be used to illustrate the execution control and data flow in each state and mode.

Tailoring Guidelines:

The states and modes, addressed by the component, should be described in this paragraph at the subcomponent level.

3.1.3 Memory and processing time allocation

This paragraph shall be numbered 3.1.3 and shall document the allocation of memory and processing time to the CSCs. The allocation may be illustrated by a memory/processing time table (see Table 1).

CSC NAME	CSC NUMBER	MEMORY BUDGET (WORDS)	ALLOCATED PROCESSING TIME
MODE CONTROL	25	1,700	128.0 ms
COORDINATE CONVERSION	69	900	155.0 ms 156.0 ms
RADAR CONTROL	26	3,000	96.0 ms
WEAPON CONTROL	27	2,100	100.0 ms
TARGET ENGAGEABILITY	11	1,700	10.0 ms
EXECUTIVE	1	1,200	80.0 ms
DATA BASE	100	2,000	N/A
TOTAL		12,600	570 ms
TOTAL AVAILABLE		16,384	740 ms
RESERVE		3,784	170 ms
RESERVE(%)		23	23

Table 3. Example of a CSC memory/processing time table

Tailoring Guidelines:

If appropiate, a preliminary allocation of memory and processing time to subcomponents can be provided in this section.

3.2 CSCI design description

This section shall be numbered 3.2 and shall be divided into the following subparagraphs to provide a design description of each CSC of the CSCI.

3.2.X (CSC name and project unique identifier)

This subparagraph shall be numbered 3.2.X (beginning with 3.2.1), shall identify a CSC by name and project unique identifier, and shall state its purpose. This subparagraph shall provide the following information:

a. Identify the requirements allocated to the CSC from the applicable requirements specification(s). If the CSC is composed of sub-level CSCs, some or all of this information may be referenced and provided by the sub-level CSC description.

b. Describe the preliminary design of the CSC in terms of execution control and data flow. If a CSC is composed of sub-level CSCs, this description shall identify the relationships among the sub-level CSCs. In addition this description shall identify each CSCI internal interface documented in the Software Requirements Specification, that is to be addressed by the CSC and its sub-level CSCs, as applicable. This information may be referenced rather than duplicated for each sub-level CSC.

c. Identify the derived design requirements for the CSC and any design constraints imposed on or by the CSC. If the CSC is composed of sub-level CSCs, some or all of this information may be referenced and provided by the sub-level CSC description.

Tailoring Guidelines:

Those requirements allocated to the CSC (component or subcomponent) from the associated Component Specification should be identified, along with a rationale for the allocation. Execution control, data flow, internal interfaces, derived design requirements, and design constraints will also be described in this subparagraph at the subcomponent level. Those subcomponents (CSCs) which will come from the reusable component repository, and those which will be newly developed, should be identified. For reusable subcomponents, reference should be made to the associated software design documentation.

3.2.X.Y (Sub-level CSC name and project unique identifier): This subparagraph shall be numbered 3.2.X.Y (beginning with 3.2.1.1), shall identify a sub-level CSC by name and project unique identifier, shall state its purpose, and shall provide the information required by a through c above. This subparagraph does not apply if there are no sub-level CSCs. If this CSC is also composed of sub-level CSCs, each sub-level CSC shall be identified by name and project unique identifier and the information required by a through c above shall be provided in a separate subparagraph for each sub-level CSC.

Tailoring Guidelines:

The same tailoring guidelines described above for Section 3.2.X are also applicable to this subparagraph.

4. Detailed design

This section shall be numbered 4 and shall be divided into the following paragraphs and subparagraphs to describe the detailed design of each CSC.

4.X (CSC name and project unique identifier)

This paragraph shall be numbered 4.X (beginning with 4.1), and shall be divided into the following subparagraphs to identify and describe each of the Computer Software Units (CSUs) of a CSC. This paragraph shall describe the relationships of the CSUs in terms of execution control and data flow between the CSUs of this CSC and shall identify all CSU interfaces that are external to the CSC. Each CSU that is used by more than one CSC shall be described in detail under one CSC and then referenced by the other using CSCs.

Tailoring Guidelines:

The CSUs described in this paragraph can represent the logical or physical organization of subcomponents (CSCs). If the CSUs represent the physical organization, then they will normally be Ada library units, i.e., package specifications and bodies, procedures, functions, or tasks (JLC89). Those CSUs which will come from the reusable component repository, and those which will be newly developed, should be identified. For reusable CSUs, reference should be made to the associated software design documentation. If appropiate, execution control, data flow, and external

interfaces will be described at the CSU level. Where used, standard interfaces should be identified and described, as required.

4.X.Y (CSU name and project unique identifier)

This subparagraph shall be numbered 4.X.Y (beginning with 4.1.1) and shall identify a CSU by name and project unique identifier and shall state the purpose of the CSU. This subparagraph shall be divided into the following subparagraphs to provide the design information for the CSU.

Tailoring Guidelines:

Those CSUs which will come from the reusable component repository, and those which will be newly developed, should be identified. For reusable CSUs, reference should be made to the associated software design documentation.

4.X.Y.1 (CSU name) Design specification/constraints: This subparagraph shall be numbered 4.X.Y.1 (beginning with 4.1.1.1) and shall state the design requirements for the CSU. This subparagraph shall identify the requirements allocated to the CSC that are to be satisfied or partially satisfied by the CSU and shall identify any constraints on the design of the CSU. The design requirements addressed in this subparagraph shall include design requirements for the man-machine interface, as applicable.

Tailoring Guidelines:

Those CSC requirements satisfied or partially satisfied by the CSU (subcomponent) should be identified, along with a rationale for the allocation. Design requirements and design constraints will also be described in this subparagraph.

4.X.Y.2 (CSU name) Design: This subparagraph shall be numbered 4.X.Y.2 (beginning with 4.1.1.2) and shall specify the design of the CSU. If the CSU is to be coded in a programming language other than the specified CSCI language, the programming language shall be identified and the rationale for its use shall be provided. If the CSU resides in a library, this subparagraph shall identify the library by name and project unique identifier, and the design document in which the library description can be found. The detailed design information identified below shall be provided for the CSU, as applicable. This information may be provided by automated tools or other techniques, such as a program design language, flowcharts, or other design representations.

a. Input/output data elements. Identify and state the purpose of each input and output data element to the CSU. The design information for data elements shall be provided in section 5.

b. Local data elements. Identify and state the purpose of each data element that originates in the CSU and is not used by any other CSU. Each data element shall be described in terms of name, brief description, data type, data representation, size, units of measure, limit/range, accuracy, precision/resolution, and any other attributes of the data. This information may be provided in a CSU local data definition table.

c. Interrupts and signals. Identify and describe the interrupts and signals handles by the CSU. Identify for each interrupt and signal, as appropriate, its source, purpose, priority, expected response and response time, and minimum, maximum, and probable frequency of occurrence.

d. Algorithms. Identify, state the purpose, and describe in detail the algorithms to be incorporated into the execution of the CSU. The algorithms shall be described in terms of the manipulation of input and local data elements and the generation of output data elements.

e. Error handling. Identify and describe the error detection and recovery features of the CSU, including handling of erroneous input data and other conditions that effect the execution of the CSU.

f. Data conversion. Identify and describe any data conversion operations performed in order to implement the CSU's interfaces.

g. Use of other elements. Describe the use of other elements that are used by the CSU including, but not limited to:

   1) Other CSUs (e.g., calls for library functions, calls for I/O services for access to databases, mass storage devices, and real-time I/O channels).

   2) Shared data stored in a global memory (e.g., databases or data files, tables, compool, datapool, etc.).

   3) Input and output buffers, including message buffers.

h. Logic flow. Describe the logic flow of the CSU in terms of items "a" through "g" above. Describe the conditions under which CSU execution is initiated and, if applicable, communication interface features are invoked, and the conditions under which control is passed to other CSUs, as applicable, If sequencing is dynamically controlled during the CSCI's operations, the method for sequence control and the logic and input conditions of that method shall be described, such as timing variations, priority assignments, internal operations such as data transfer in and out of internal memory, sensing of discrete input signals, and timing relationships between interrupt operations with the CSCI.

i. Data structures. Describe local data structures implemented by the CSU and any shared data structures used by the CSU. Shared data structures shall be described under one CSU and referenced thereafter by the sharing CSUs.

j. Local data files or database. If a data file(s) or a database are part of the local data of a CSU, state the purpose of each file or database, the structure of each file or database in terms of records, fields, etc., and describe the access procedures, such as sequential or random.

k. Limitations. Describe any limitations or unusual features that restrict the performance of the CSU.

Tailoring Guidelines:

For those CSUs which will come from the reusable component repository, the repository should be identified and reference made to the associated software design documentation. For those which will be newly developed, design information in this paragraph should be as defined in the DID instructions above, and at a sufficient level of detail to allow the CSU to be implemented. For a project using OOD, the input/output data elements would describe the messages, i.e., the operations performed by an object, and the responses to those messages. The local data files or database would describe the encapsulated data of an object.

5. CSCI data

This section shall be numbered 5 and shall describe the global data elements within the CSCI. For ease in readability and maintenance, the information required below may be provided in one or more tables. The following information shall be provided for each data element, as applicable:

a. For data elements internal to the CSCI:

   1) Name of the data element

   2) A brief description

   3) The units of measure, such as knots, seconds, meters, feet, etc.

   4) The limit/range of values required for the data element (for constants provide the actual value)

   5) The accuracy required for the data element

   6) The precision/resolution in terms of significant digits

7) For real time systems, the frequency at which the data element is calculated or refreshed such as 10 KHz, 50 Msec, etc.

8) Legality checks performed on the data element

9) The data type, such as integer, ASCII, fixed, real, enumeration, etc.

10) The data representation/format

11) The CSU project unique identifier where the data element is set or calculated

12) The CSU project unique identifier(s) where the data element is used

13) The data source from which the data is supplied, such as database or data file, global common, local common, compool, datapool, parameter, etc. Where applicable, each source shall be identified by its project unique identifier.

b. For data elements of the CSCI's external interfaces:

1) Identify the data element

2) Identify the interface by name and project unique identifier

3) Reference the Interface Design Document (IDD) in which the external interface is described.

Tailoring Guidelines:

This subparagraph will describe data elements internal and external to this component. The data element descriptions should contain sufficient detail to support the implementation of the component.

6. CSCI data files

This section shall be numbered 6 and shall be divided into the following paragraphs to describe each of the shared data files of the CSCI.

6.1 Data file to CSC/CSU cross reference

This paragraph shall be numbered 6.1 and shall provide a mapping of each data file identified below to the CSCs and CSUs that use the data file.

Tailoring Guidelines:

This paragraph will provide a mapping of the data files to the subcomponents that use the data file. For a project using OOD, the data files represent the encapsulated data of the component objects.

6.X (Data file name and project unique identifier)

This subparagraph shall be numbered 6.X (beginning with 6.2) and shall identify by name and project unique identifier a data file of the CSCI that is shared by more than one CSU. This paragraph shall state the purpose of the data file, identify the maximum size of the file, and describe the file access method, such as random or sequential. This paragraph shall provide a description of the structure and size of the records contained within the file. This paragraph shall also provide a description of the data that is to reside in the file. The data description shall include, as applicable, data type, data representation, size, units of measure, limit/range, accuracy, precision/resolution, and any other design characteristics of the data. This information may be provided in a file definition table.

Tailoring Guidelines:

For a project using OOD, the data files represent the encapsulated data of the component objects. For encapsulated data within an object that is shared by more than one CSU, this paragraph should describe the data at a sufficient level of detail to support the implementation of the object.

7. Requirements traceability

This section shall be numbered 7 and shall provide traceability of the requirements allocated down to the CSU level of each CSC back to the requirements of the Software Requirements Specification and Interface Requirements Specification. The traceability may be shown graphically.

Tailoring Guidelines:

Requirements at the CSU level of each CSC will be traced back to the requirements in the Component Specification.

DoD-STD-2167A DID: Interface Design Document

IDENTIFICATION NUMBER: DI-MCCR-80027A

3. Interface design

This section shall be numbered 3 and shall be divided into the following paragraphs and subparagraphs to describe the interface design for those interfaces to which this IDD applies.

3.1 Interface diagrams

This paragraph shall be numbered 3.1 and shall specify for each CSCI to which this IDD applies, its relationship to the other HWCIs, CSCIs, or critical items with which it interfaces. This description may be provided by one or more interface diagrams, as appropriate.

Tailoring Guidelines:

This paragraph will show the relationships of this component to other hardware or software components to which it interfaces. Some of these hardware or software components may be prototypes which are part of a system prototype. Where used, standard interfaces should be identified.

3.X (Interface name and project-unique identifier)

This paragraph shall be numbered 3.X (beginning with 3.2), shall identify an interface by name and project-unique identifier, and shall state its purpose. This paragraph shall be divided into the following subparagraphs to describe the design of the interface.

Tailoring Guidelines:

This paragraph, and the subparagrahs that follow, shall provide the design of the component interfaces (the level of detail will be project dependent). Where used, standard interfaces should be identified and the associated software design documentation referenced.

3.X.1 Data elements

This subparagraph shall be numbered 3.X.1 (beginning with 3.2.1) and shall provide, in a data element definition table, the following information, as applicable, for each data element transmitted across the interface:

a. A project-unique identifier for the data element

b. A brief description of the data element

c. The CSCI, HWCI, or critical item that is the source of the data element

d. The CSCI(s), HWCI(s), or critical item(s) that are the users of the data element

e. The units of measure required for the data element, such as seconds, meters, kilohertz, etc.

f. The limit/range of values required for the data element (for constants provide the actual value)

g. The accuracy required for the data element

h. The precision or resolution required for the data element in terms of significant digits.

i. The frequency at which the data element is calculated or refreshed, such as 10 KHz or 50 Msec

j. Legality checks performed on the data element

k. The data type, such as integer, ASCII, fixed, real, enumerated, etc.

l. The data representation/format

m. The priority of the data element.

3.X.2 Message descriptions

This subparagraph shall be numbered 3.X.2 (beginning with 3.2.2), shall identify each message transmitted across the interface by name and project-unique identifier, and shall describe the assignment of data elements to each message. A cross-reference of each message to the data elements that embody the message shall be provided. In addition, a cross-reference of each data element to the message(s) of which it is a part shall also be provided. Cross-references may be provided as an appendix and referenced in this subparagraph.

3.X.3 Interface priority

This subparagraph shall be numbered 3.X.3 (beginning with 3.2.3) and shall specify the relative priority of the interface and of each message transmitted across the interface.

3.X.4 Communications protocol

This subparagraph shall be numbered 3.X.4 (Beginning with 3.2.4) and shall be divided into the following subparagraphs to describe the commercial, military, or proprietary communications protocols associated with the interface.

3.X.4.Y (Protocol name): This subparagraph shall be numbered 3.X.4.Y (beginning with 3.2.4.1), shall identify a protocol by name and shall describe the technical details of the protocol. This subparagraph shall address the following communications specification details, as applicable:

a. Fragmentation and reassembly of messages

b. Message formatting

c. Error control and recovery procedures, including fault tolerance features

d. Synchronization, including connection establishment, maintenance, termination, and timing

e. Flow control, concluding sequence numbering, window size, and buffer allocation

f. Data transfer rate, whether it is periodic or aperiodic, and minimum interval between transfers

g. Routing, addressing, and naming conventions

h. Transmission services, including priority and grade

i. Status, identification, notification, and any other reporting features

j. Security, including encryption, user authentication, compartmentalization and auditing.

Appendix I. Software Test Document

This appendix describes how the DoD-STD-2167A CDRL item DID (Data Item Description), for the Software Test Plan (STP), Software Test Description (STD), and Software Test Report (STR), can be tailored for use as the SFLC Component Software Test Document (SWTD).

Differences Between DoD-STD-2167A and SFLC:

The Software Test Plan (STP), as defined in STP DID, "describes the formal test plans for one or more Computer Software Configuration Items (CSCIs)." The Software Test Description (STD), as defined in STD DID, "contains the test cases and test procedures necessary to perform formal qualification testing of a Computer Software Configuration Item (CSCI) identified in the Software Test Plan (STP)." The Software Test Report (STR), as defined in STD DID, "is a record of the formal qualification testing performed on a Computer Software Configuration Item (CSCI)." On a typical DoD-STD-2167A project, a single STP describes the overall test plan and environment for all the CSCIs in the system, while individual STDs and STRs are used to describe detailed test requirements and results for each CSCI.

In the SFLC, there will be a single system-level Software Test Document (SWTD) and multiple component-level SWTDs. All of them will be evolving documents. In the System Architecture phase, the system-level SWTD will expand, as needed, as the system prototypes evolve into a full-capability prototype. In the Productization and Production phase, where the full-capability prototype is converted into a productized system, the system-level SWTD will expand to include the system testing required of the productized system. In the Software Growing phase, each component-level SWTD will expand, as needed, as the associated component prototype evolves into a full-capability prototype, and then into a productized component.

General Tailoring Guidelines:

SWTDs will provide planning and documentation of testing of the system and component protoypes, as well as the productized system and its associated components. For prototypes, the SWTDs need to describe the demonstration and evaluation of the prototypes, including evaluation criteria and procedures. As mentioned above, all of the SWTDs will be evolving documents and will contain the information defined in the STP, STD, and STR DIDs. CSCIs, identified in these DIDs, will be either closely related sets of software components and/or individual high level software components. The choices will, of course, depend on the specific application. For reusable components, associated SWTDs can be referenced, as needed.

Specific Tailoring Guidelines:

The DID text for each section of the STP, STD, and STR are provided below. No specific section-level tailoring guidelines are included with the DID text. For a given project, the above general tailoring guidelines would be applied, as appropiate, to tailor each section for use in a system-level or component-level SWTD. Other tailoring will be needed, to make it reflect the unique requirements of the associated system or component, and the specific software development methodologies being used.

DoD-STD-2167A DID: Software Test Plan

IDENTIFICATION NUMBER: DI-MCCR-80014A

3. Software test environment

This section shall be numbered 3 and shall be divided into the following paragraphs to identify and describe the plans for implementing and controlling the resources (software, firmware, and hardware) necessary to perform formal qualification testing. To reduce duplication, references may be made in the paragraphs below to the software engineering environment described in the Software Development Plan (SDP) for those resources that are used in both environments.

3.1 Software items

This paragraph shall be numbered 3.1 and shall identify the software items (e.g., operating systems, compilers, code auditors, dynamic path analyzers, test drivers, preprocessors, test data generators, post-processors) necessary to perform the formal qualification testing activities. This paragraph shall describe the purpose of each item and shall identify any classified processing or security issues associated with the software items.

3.2 Hardware and firmware items

This paragraph shall be numbered 3.2 and shall identify the computer hardware, interfacing equipment, and firmware items that will be used in the software test environment. This paragraph shall describe the purpose of each item and shall identify any classified processing or security issues associated with the hardware or firmware items.

3.3 Proprietary nature, and Government rights

This paragraph shall be numbered 3.3 and shall identify the proprietary nature and Government rights associated with each item of the software test environment.

3.4 Installation, testing and control

This paragraph shall be numbered 3.4 and shall identify the contractor's plans for installing and testing each item prior to its use. This paragraph shall also describe the contractor's plans for controlling and maintaining each item of the software test environment.

4. Formal qualification test identification

This section shall be numbered 4 and shall be divided into the following paragraphs and subparagraphs to identify each formal qualification test and to describe the formal qualification test requirements for each CSCI to which this STP applies.

4.X (CSCI name and project-unique identifier)

This paragraph shall be numbered 4.X (beginning with 4.1), shall identify a CSCI by name and project-unique identifier, and shall be divided into the following subparagraphs to describe the total scope of testing for the CSCI.

4.X.1 General test requirements

This subparagraph shall be numbered 4.X.1 (beginning with 4.1.1) and shall describe requirements that apply to all of the formal qualification tests or to a group of formal qualification tests. For example:

"Each formal qualification test shall meet the following general test requirements":

a. CSCI size and execution time shall be measured.

b. The CSCI shall be tested using nominal, maximum, and erroneous input values.

c. The CSCI shall be tested for error detection and proper error recovery, including appropriate error messages.

"Formal qualification tests to validate radar tracking requirements shall meet the following test requirements":

a. The CSCI shall be tested using simulated test data for the specified combinations of environmental conditions.

b. The CSCI shall be tested using input data taken from the environment ("live data")

4.X.2 Test classes

This paragraph shall be numbered 4.X.2 (beginning with 4.1.2) and shall describe the types or classes of formal qualification tests that shall be executed (e.g., stress tests, timing tests, erroneous input tests, maximum capacity tests).

4.X.3 Test levels

This paragraph shall be numbered 4.X.3 (beginning with 4.1.3) and shall describe the levels at which formal qualification testing will be performed. For example:

a. CSCI level (CSC or CSU level if necessary) - to evaluate compliance with CSCI requirements.

b. CSCI to CSCI integration level - to evaluate compliance with CSCI external interface requirements.

c. CSCI to HWCI integration level - to evaluate compliance with CSCI external interface requirements.

d. System level - to evaluate compliance with CCCI requirements not evaluated at other levels.

4.X.4 Test definitions

This paragraph shall be numbered 4.X.4 (beginning with 4.1.4) and shall be divided into the following subparagraph to identify and describe each formal qualification test to be conducted on the CSCI.

4.X.4.Y (Test name and project-unique identifier): This subparagraph shall be numbered 4.X.4.Y (beginning with 4.1.4.1) and shall identify a formal qualification test by name and project-unique identifier. This subparagraph shall provide the information specified below for the test. Some or all of this information may be provided graphically.

a. Test objective

b. Any special requirements (e.g., 48 hours of continuous facility time, weapon simulation)

c. Test level

d. Test type or class

e. Qualification method as specified in the Software Requirements Specification

f. Cross reference to the CSCI engineering requirements in the Software Requirements Specification addressed by this test.

g. Cross reference to the CSCI interface requirements in the Interface Requirements Specification addressed by this test.

h. Type of data to be recorded

i. Assumptions and constraints.

4.X.5 Test schedule

This paragraph shall be numbered 4.X.5 and shall contain or reference the test schedule for conducting the tests identified in paragraph 4.X.4.

5. Data recording, reduction, and analysis

This section shall be numbered 5 and shall be divided into paragraphs and subparagraphs as appropriate to describe the data reduction and analysis procedures to be used during and following the tests identified in this STP. This section shall document how information resulting from data reduction and analysis will be retained. The results of data recording, reduction, and analysis activities shall be documented in such a way that the resulting information will clearly show whether the test objectives have been met.

DoD-STD-2167A DID: Software Test Description

IDENTIFICATION NUMBER: DI-MCCR-80015A

3. Formal Qualification test preparations

This section shall be numbered 3 and shall be divided into the following paragraphs and subparagraphs to describe the test schedule and pre-test procedures for each formal qualification test of the CSCI as identified in the Software Test Plan (STP).

3.X (Test name and project-unique identifier)

This paragraph shall be numbered 3.X (beginning with 3.1), shall identify a formal qualification test by name and identifier, and shall be divided into the following subparagraphs to describe the test schedule and pre-test schedule and pre-test procedures for the test.

3.X.1 (Test name) schedule

This subparagraph shall be numbered 3.X.1 (beginning with 3.1.1) and shall provide (either directly or by reference) the location and schedule for the following activities associated with the test, as applicable:

a. Briefings

b. Pre-test activities (e.g., equipment and software preparation)

c. Test

d. Debriefings

e. Data reduction and analysis.

3.X.2 (Test name) pre-test procedures

This subparagraph shall be numbered 3.X.2 and shall be divided into the following subparagraphs to describe the preparation and setup for the test. When the information required duplicates information previously specified for another test, that information may be referenced rather than repeated.

3.X.2.1 Hardware preparation: This subparagraph shall be numbered 3.X.2.1 and shall describe the procedures necessary to prepare the hardware for the test. Reference may be made to published operating manuals for these procedures. The following shall be provided, as applicable:

a. The specific hardware to be used, identified by name and, if applicable, number.

b. Any switch settings and cabling necessary to connect the hardware. Those shall be identified by name and location.

c. One or more diagrams to show hardware, interconnecting control, and data paths.

d. Precise step-by-step instructions for placing the hardware in a state of readiness.

3.X.2.2 Software preparation: This subparagraph shall be numbered 3.X.2.2 and shall describe the procedures and related information necessary to prepare the CSCI and support software for the test. Reference may be made to published software manuals for these procedures. The following information shall be provided, as applicable:

a. The storage medium of the CSCI (e.g., magnetic tape, diskette) and precise step-by-step instructions for loading the CSCI into the computer.

b. The storage medium of any support software (e.g., environment simulators, test drivers, data reduction programs, etc.) and precise step-by-step instruction for loading the support software.

c. When the support software is to be loaded (e.g., before the CSCI has been loaded, after a specific test case has been executed, etc.).

d. The instructions for CSCI and support software initialization that are common to more than one test case.

3.X.2.3 Other pre-test preparations: This subparagraph shall be numbered 3.X.2.3 and shall describe any other pre-test preparations or procedures necessary to perform the test.

4. Formal qualification test descriptions

This section shall be numbered 4 and shall be divided into the following paragraphs and subparagraphs to identify the test cases, test procedures, and related information associated with each formal qualification test of the CSCI identified in the Software Test Plan (STP).

4.X (Test name and project-unique identifier)

This paragraph shall be numbered 4.X (beginning with 4.1) and shall identify a formal qualification test by name and project-unique identifier. The following subparagraphs shall provide the detailed test description for the test.

4.X.Y (Test case name and project-unique identifier)

This subparagraph shall be numbered 4.X.Y (beginning with 4.1.1), shall identify a test case by name and project-unique identifier, state its purpose, and provide a brief description. The following subparagraphs shall provide a detailed description of the test case.

4.X.Y.1 (Test case name) requirements traceability: This subparagraph shall be numbered 4.X.Y.1 (beginning with 4.1.1.1) and shall identify the engineering requirements in the Software Requirements Specification and the interface requirements in the interface Requirements Specification that are addressed by the test case.

4.X.Y.2 (Test case name) initialization: This subparagraph shall be numbered 4.X.Y.2 and shall identify any prerequisite conditions that must be established prior to performing the test case. When the information required in this subparagraph duplicates information previously specified, that information may be referenced rather than repeated. The following considerations shall be discussed, as applicable:

a. Hardware and software configuration

b. Flags, initial breakpoints, pointers, control parameters, or initial data to be set/reset prior to test commencement

c. Preset hardware conditions or electrical states necessary to run the test case.

d. Initial conditions to be used in making timing measurements

e. Conditioning of the simulated environment

f. Special instructions peculiar to the test case.

4.X.Y.3 (Test case name) test inputs: This subparagraph shall be numbered 4.X.Y.3 and shall describe the test inputs necessary for the test case. The following shall be provided, as applicable:

a. Name, purpose, and description (e.g., range of values, accuracy, etc.) of each test input

b. Source of the test input and the method to be used for selecting the test input

c. Whether the test input is real or simulated

d. Time or event sequence of the test input (e.g., in response to question Y, input 4.375).

4.X.Y.4 (Test case name) expected test results: This subparagraph shall be numbered 4.X.Y.4 and shall identify all expected test results for the test case. Both intermediate and final test results shall be provided, as applicable.

4.X.Y.5 (Test case name) criteria for evaluation results: This subparagraph shall be numbered 4.X.Y.5 and shall identify the criteria to be used for evaluating the final results of the test case. When the information required in this subparagraph duplicates information previously specified, that information may be referenced in this subparagraph. For each test result, the following information shall be provided, as applicable:

a. Accuracy requirements for the test result

b. Allowable upper and lower bounds of the test result

c. Maximum and minimum duration of the test, in terms of time or number of events, in order to obtain the test result

d. Conditions under which the test result is inconclusive and re-testing is to be performed

e. Severity of processing errors associated with the test result

f. Additional criteria not mentioned above.

4.X.Y.6 (Test case name) test procedure: This subparagraph shall be numbered 4.X.Y.6 and shall define the test procedure for the test case. The test procedure shall be defined as a series of individually numbered steps listed sequentially in the order in which the steps are to be performed. For convenience in document maintenance, the lost procedure may be included as an appendix and referenced in this subparagraph. The following shall be provided for each test procedure, as applicable:

a. Test operator actions and equipment operation required for each step

b. Expected result for each step

c. Evaluation criteria for each step, as applicable

d. Actions to follow in the event of a program stop or indicated error

e. Procedures to be used to reduce and analyze test results.

4.X.Y.7 (Test case name) assumptions and constraints: This subparagraph shall be numbered 4.X.Y.7 and shall identify any assumptions made and constraints imposed in the description of the test case. If waivers or exceptions to specified limits and parameters are approved, they shall be identified and this subparagraph shall address their effects and impacts upon the test case.

DoD-STD-2167A DID: Software Test Report

IDENTIFICATION NUMBER: DI-MCCR-80017A

3. Test Overview

This section shall numbered 3 and shall be divided into the following paragraphs and subparagraphs to describe the results of each formal qualification test covered by this report.

3.X (Formal qualification test name and project-unique identifier)

This paragraph shall be numbered 3.X (beginning with 3.1), shall identify a formal qualification test by name and number, and shall be divided into the following subparagraphs to provide an overview of the test results.

3.X.1 (Formal qualification test name) summary

This subparagraph shall be numbered 3.X.1 and shall summarize the results of the formal qualification test. The summary shall include the completion status of the test (i.e., success or failure). For failures, the step of the test procedure where the failure occurred and an identification of the resulting problem report shall be included. This information may be provided by reference to a test results summary table similar to Table I.

3.X.2 (Formal qualification test name) test record

This paragraph shall be numbered 3.X.2 and shall present a chronological record of all events relevant to test preparation, test performance, and analysis and interpretation of formal qualification test results. This paragraph may reference a test log that contains the chronological record of the conduct of the formal qualification test. This subparagraph shall include the following information:

a. The date(s), time(s) and location(s) of the test, as well as hardware and software configurations used for the test. The description of the test configuration shall include, when available, part number, model number, serial number, manufacturer, revision level, and

calibration date of all hardware, and version number and name for the software components used.

b. The date and time of each test-related activity, the identity of the individual(s) who performed the activity, and the identities of witnesses.

c. Any problems encountered and the specific step(s) of the test procedures associated with the problem, including the number of times an individual step in a procedure had to be repeated in attempting to correct a problem and the outcome of each attempt.

d. Back-up points or test steps where tests were resumed for retesting.

4. Test results

This section shall be numbered 4 and shall be divided into the following paragraphs to describe the detailed results for each formal qualification test.

4.X(Formal qualification test name & project unique ID) test results

This paragraph shall be numbered 4.X (beginning with 4.1), shall identify a formal qualification test by name and project-unique identifier, and shall be divided into the following subparagraphs to describe the detailed results for each test case of the formal qualification test.

4.X.Y (Test case name and project-unique identifier)

This subparagraph shall be numbered 4.X.Y (beginning with 4.1.1), shall identify a test case by name and project-unique identifier, and shall be divided into the following subparagraphs to describe the detailed results for the test case.

4.X.Y.1 (Test case name) test results: This subparagraph shall be numbered 4.X.Y.1 and shall present the test results for the test case. For each step of the test procedure executed, the result shall be recorded. Any anomalies or discrepancies of any kind encountered during the execution of the test case shall be described in this subparagraph. Amplifying information (e.g., memory dumps, record of registers, display diagrams) that may help to isolate and correct the cause of any discrepancies shall be included or referenced. The assessment of the test conductor as to the cause of each discrepancy and a means of correcting it may be provided.

4.X.Y.2 (Test case name) deviations from test procedure: This subparagraph shall be numbered 4.X.Y.2 and shall discuss in detail any deviations from the test procedure described in the corresponding Software Test Description (e.g., substitution of required equipment, changes to support software, procedural steps not followed, and schedule deviations). For each deviation, the rationale for allowing it and its impact on the validity of the test shall be provided.

5. CSCI evaluation and recommendations

This section shall be numbered 5 and shall be divided into the following paragraphs.

5.1 CSCI evaluation

This paragraph shall be numbered 5.1 and shall provide an overall analysis of the capabilities of the CSCI demonstrated by the test results in this report. The analysis shall identify any remaining deficiencies, limitations, or constraints in the CSCI that were detected by the testing performed. Software problem/change reports may be used to provide deficiency information. For each deficiency, limitation, or constraint, the analysis shall:

a. Describe its impact on CSCI and system performance

b. Describe the impact on the CSCI and system design in order to correct it

c. Provide a recommended solution/approach for correcting it.

5.2 Recommended improvements

This paragraph shall be numbered 5.2 and shall provide any recommended improvements in the design, operation, or testing of the CSCI. A discussion of each recommendation and its impact on the CSCI may be provided. If no recommended improvements are provided, this paragraph shall state "None".